# Right Ho, T24

By V.Kazimirchik, last update:  Sun, 05 Apr 2015 11:45 UTC

*Technical insides of Temenos T24 - revisited, retested, expanded, updated and enhanced.*

# Contents

# 1 What's what

## 1.1 What is T24, jBASE, TAFC, jsh, jBC, jQL

• T24 is a core banking system written mostly in jBC language (also called Infobasic). Compiled first to C (TAFC) and then to binaries/libraries for corresponding platform. TAFJ is not so widely used alternative to TAFC that uses Java as an intermediate source code.

• jBASE is post-relational DBMS (also known as multi-valued database).

• TAFC stands for "Temenos application framework for C" - usually packed together with jBASE.

• jsh - "jBASE shell" which is normally launched when one logs in to T24 environment via terminal emulator program like Putty or SecureCRT.

• jQL is query language seemingly similar to SQL but quite different in many aspects.

• T24 "Classic" is T24 text user interface which is launched from jsh.

• T24 Browser is graphical user interface that requires only Web browser on the client side.

## 1.2 jBC features

• Possibility to work with hashed or sequential files.

• "Hashed" mode can work transparently with hashed files, UD and SQL types - see "jBASE features" below.

• Local and global variables.

• Case sensitivity for variables and keywords.

• Weak variables typing; implicit types conversion.

• Dynamic and dimensioned arrays (elements numbering starts from 1 rather than from 0 like in other programming languages).

• Aliases for variables or expressions that are resolved at compile time (EQUATE).

• Manipulations with date, time, strings, numbers.

• Optional statement labels.

• Code sections accessed by GOSUB.

• (Unfortunately) GOTO clause.

• Code INSERTs.

• IF...THEN...ELSE conditional statement.

• BEGIN CASE...CASE...END CASE conditional statement.

• FOR...NEXT loop.

• LOOP...DO...WHILE/UNTIL...REPEAT loop.

• External programs or commands execution (with data, output and select lists passing/capturing).

• Patterns matching.

• Unary operators (like i++).

- Single- and multi-byte character sets support.
- Operations with SELECT lists.
- Manipulations with environment variables.
- Possibility to call Java classes or C code.
- Conditional compilation.
- Emulation modes for compatibility with numerous multi-value platforms (for T24 “prime” emulation is used).

## 1.3 Variables visibility and lifetime

Variable is visible throughout the program or a subroutine (i.e. in the bounds of particular source code file since it’s not possible to place a program and a subroutine or several subroutines into the same file).

To share a variable between different programs/subroutines pass them as parameters in CALL statement or use a named or unnamed COMMON (i.e. global variables area).

PASSDATA and COLLECTDATA statements can also be used for that.

All variables (except ones in COMMON areas) are reset (i.e. become unassigned) when program terminates.

## 1.4 jBASE features

- Different hashed file types support.
- Hashed data files distribution.
- SQL DBs for XML data storage (DB2/Oracle/MSSQL).
- UD type (directory is a table, file name is @ID, file contents is a record; each line is a field).
- Indexes.
- Triggers.
- Transactions support.
- Source code editor - JED (also capable of editing data files).
- Command-line debugger.

## 1.5 How to get there

Log in via telnet or ssh. Alternatively, on a local machine run remote.cmd or .profile (depending on platform). If all is correct, find youself in bnk.run directory of so-called “T24 environment” at jsh prompt.

## 1.6 What is “T24 environment”

Tree of directories on T24 application server that contain setup scripts, executables, system and hashed/UD data files for particular T24 instance (e.g., TEST environment, UAT environment etc). Different T24 environments can be totally in-

dependent or might share something - e.g.  TAFC or T24 core libraries.  TAFC is not considered to be a part of T24 environment and is usually installed to some different place on application server.

## 1.7 See settings

jsh

```
jdiag
```

jdiag typical output

```
       jdiag - jBASE diagnostic '$Revision: 1.15 $'

System Information
==================

System                      : WinNT R14VM 6.1 AMD64
OS Release                  : Windows 7, Build 7601, Service Pack 1
NT User                     : r14
Time                        : Tue Oct 07 19:33:51 2014

Environment
===========

JBCPORTNO                   : Not Set
TAFC_HOME                   : 'D:\Temenos\ModelBank-R14\T24\Env\..\Programs\TAFC'
WARNING: JBCGLOBALDIR is not set, Default 'D:\Temenos\ModelBank-R14\T24\Env\..\
         Programs\TAFC'
WARNING: JBCDATADIR is not set, Default 'D:\Temenos\ModelBank-R14\T24\Env\..\Programs
         \TAFC\jbase_data'
WARNING: JBCDATADIR is subdirectory of JBCGLOBALDIR
HOME                        : 'D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run'
JEDIFILENAME_MD             : Not Set
JEDIFILENAME_SYSTEM         : 'D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\SYSTEM'
RELEASE Information         : Major 14.0 , Minor 0.0 , Patch  (Change 114406)
Spooler dir (Default)       : 'D:\Temenos\ModelBank-R14\T24\Env\..\Programs\TAFC\
         jbase_data\jspooler'
WARNING: Cannot access Spooler directory 'D:\Temenos\ModelBank-R14\T24\Env\..\
         Programs\TAFC\jbase_data\jspooler', error 2
JBCEMULATE                  : 'prime'
TEMP file path              : 'C:\Windows\TEMP\'
Object path (Default)       : 'D:\Temenos\ModelBank-R14\T24\Env\..\Programs\TAFC\lib;
         D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\lib'
WARNING: From checking the registry, It appears that VC++ is not loaded
jBASE Compiler Run-time     : 'D:\Temenos\ModelBank-R14\T24\Env\..\Programs\TAFC\
         config\system.properties'
Program dir (Default)       : 'D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\bin'
Subroutine dir (Default)    : 'D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\lib'
```

## 1.8 See some data

Main T24 system table (contains one record):

jsh

```
LIST F.SPF
```

Output

```
@ID.................. SYSTEM
SYSTEM.SPEC.......... SYSTEM
RUN.DATE............. 20140328
SITE.NAME............ Model Bank R14
OP.MODE.............. O
OP.CONSOLE........... OFF
MAIN.ACCOUNT......... ../bnk.data
BACKUP.CYCLE.1.......
BACKUP.CYCLE.2.......
CURRENT.RELEASE...... R14
...
```

(*Press Q to stop the long output.*)

Branches list (IDs only):

jsh

```
LIST ONLY F.COMPANY
     ^^^^
```

Output

```
@ID........

GB0010003
GB0010005
SG0010001
EU0010001
GB0010002
GB0010004
GB0010001

 7 Records Listed
```

Same with sorting:

jsh

```
LIST ONLY F.COMPANY BY @ID
                    ^^^^^^
```

Output

```
@ID........

EU0010001
GB0010001
GB0010002
GB0010003
GB0010004
GB0010005
SG0010001

 7 Records Listed
```

“Shortcuts” that can be used in T24 instead of @ID are called “mnemonics” and are stored in a promptly named field MNEMONIC:

jsh

```
LIST F.COMPANY MNEMONIC
```

Output

```
@ID........    MNEMONIC

  GB0010003    MF2
  GB0010005    BR1
  SG0010001    SG1
  EU0010001    EU1
  GB0010002    MF1
  GB0010004    MF3
  GB0010001    BNK
```

Several records can be chosen for listing - see 2 standard (pre-defined) T24 users:

jsh

```
LIST F.USER 'INPUTTER' 'AUTHORISER'
```

Output

```
@ID.................. INPUTTER
USER.ID.............. INPUTTER
USER.NAME........... INPUTTER
SIGN.ON.NAME........ INPUTT
CLASSIFICATION...... INT
LANGUAGE............ 1
COMPANY.CODE........ GB0010001   GB0010002   EU0010001   SG0010001   GB0010005
                    GB0010003   GB0010004
DEPARTMENT.CODE..... 1
PASSWORD.VALIDITY... 20141101M0601
START.DATE.PROFILE.. 20140515
END.DATE.PROFILE.... 20160920
...
```

Note quotes in record @IDs - otherwise jQL will try to interpret them as field names:

jsh

```
LIST F.USER INPUTTER AUTHORISER
```

Output

```
@ID........ BUILDUSER98
INPUTTER... 27173_OFFICER__OFS_SEAT
AUTHORISER. 27173_OFFICER_OFS_SEAT

@ID........ BUILDUSER40
INPUTTER... 27173_OFFICER__OFS_SEAT
AUTHORISER. 27173_OFFICER_OFS_SEAT

@ID........ BUILDUSER8200
INPUTTER... 27173_OFFICER__OFS_SEAT
AUTHORISER. 27173_OFFICER_OFS_SEAT
...
```

## 1.9 Log in to T24

Firstly set up Putty (assuming that it’s a modified one with T24 keys option available).

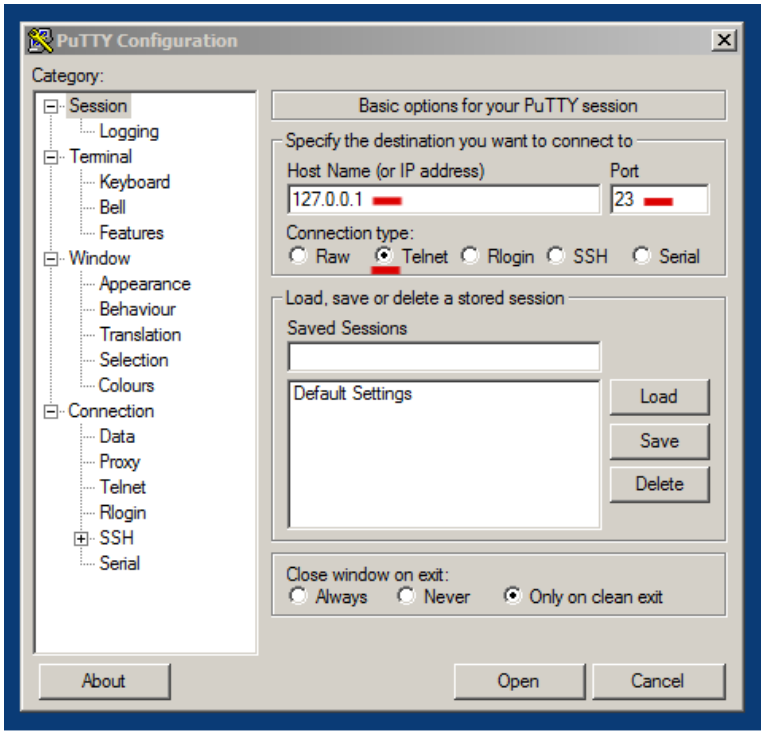


**Figure 1: Set up server IP address and connection mode/port.**

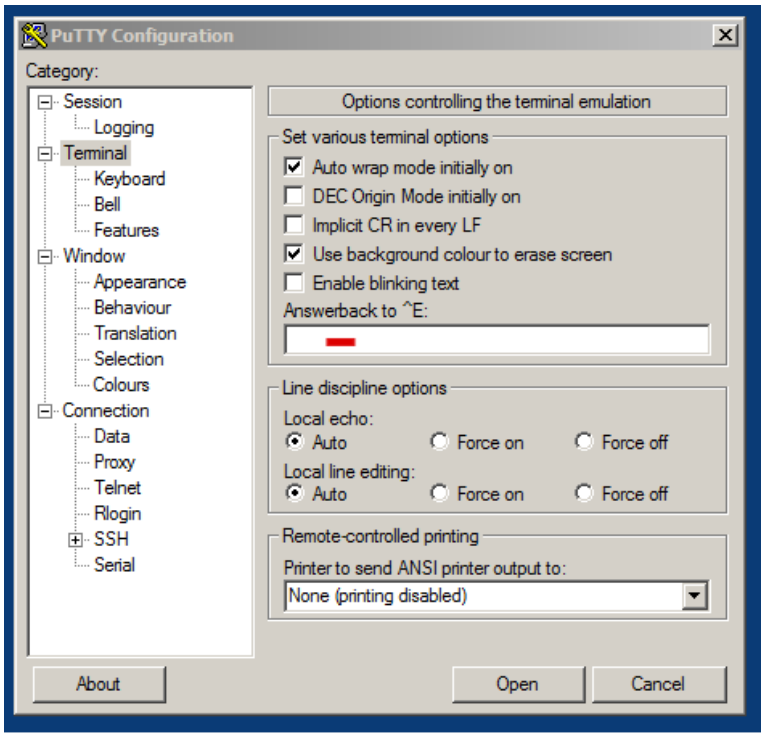


**Figure 2: Answerback to Ctrl-E should be empty.**

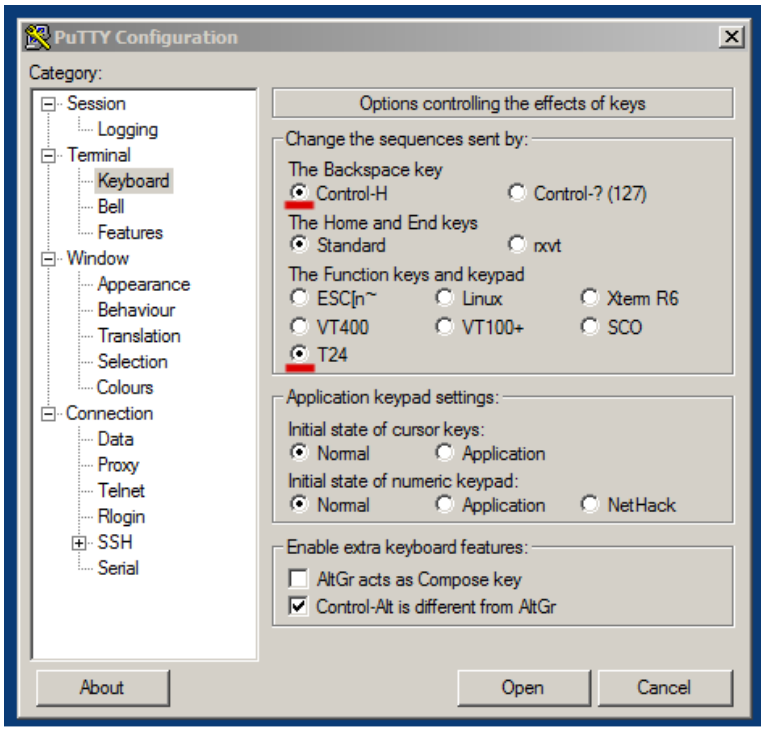


**Figure 3: Backspace key: Ctrl-H. The Function keys and keypad: T24.**

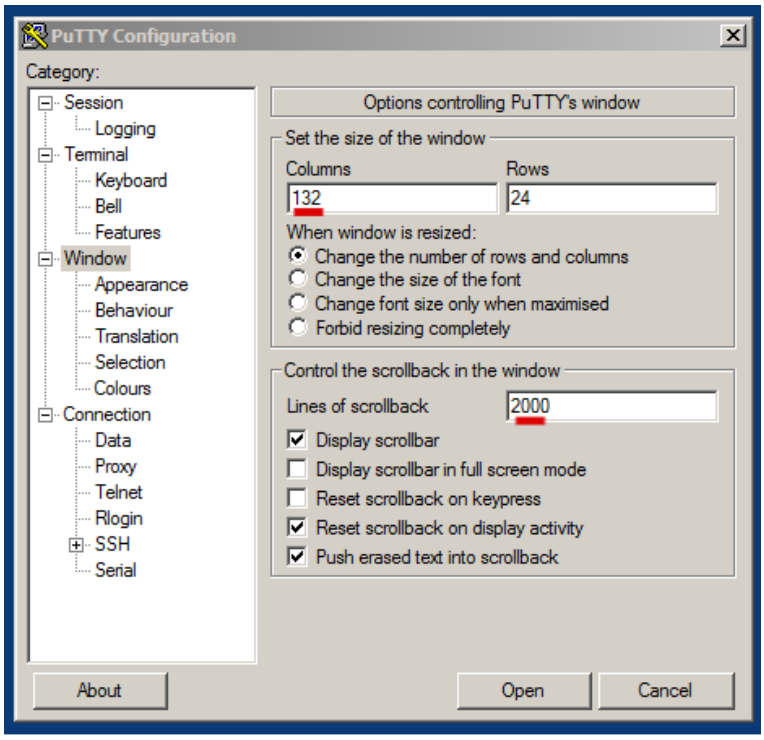


**Figure 4: Number of columns: 132, scrollback: 2000.**

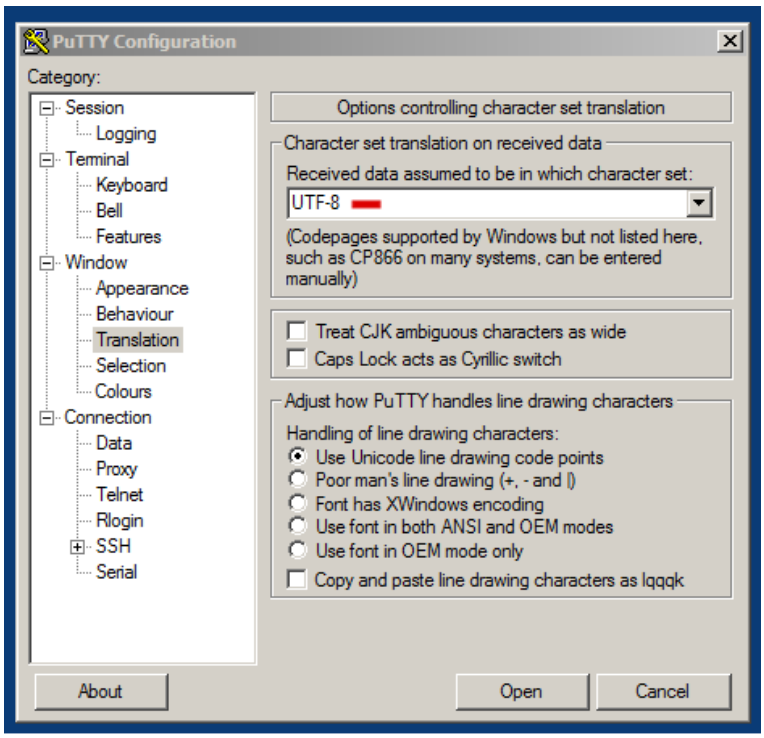


**Figure 5: Translation as per regional settings.**

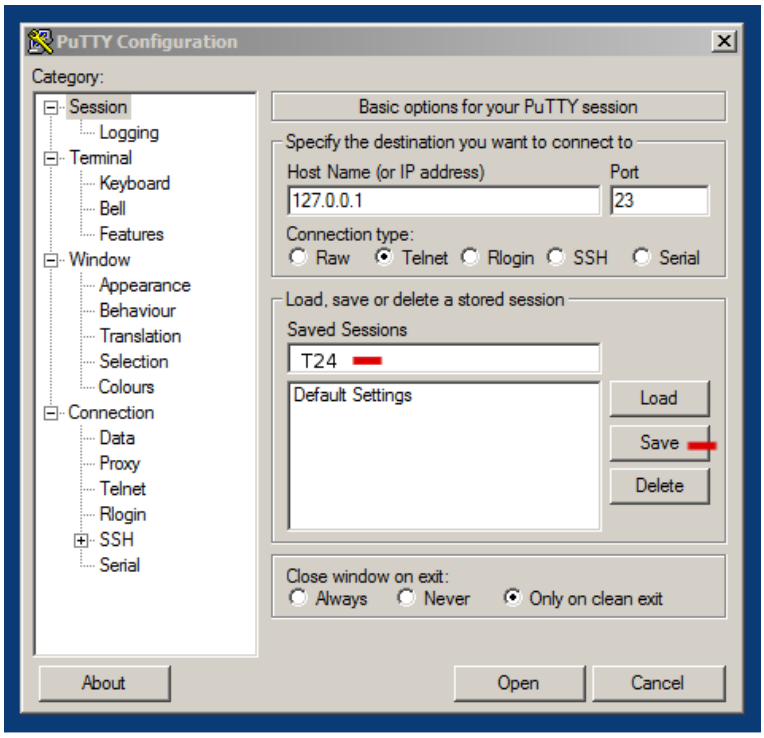


**Figure 6: Type session name and save.**

Next time when you launch Putty just double-click on a saved session (or select it and click “Open”):

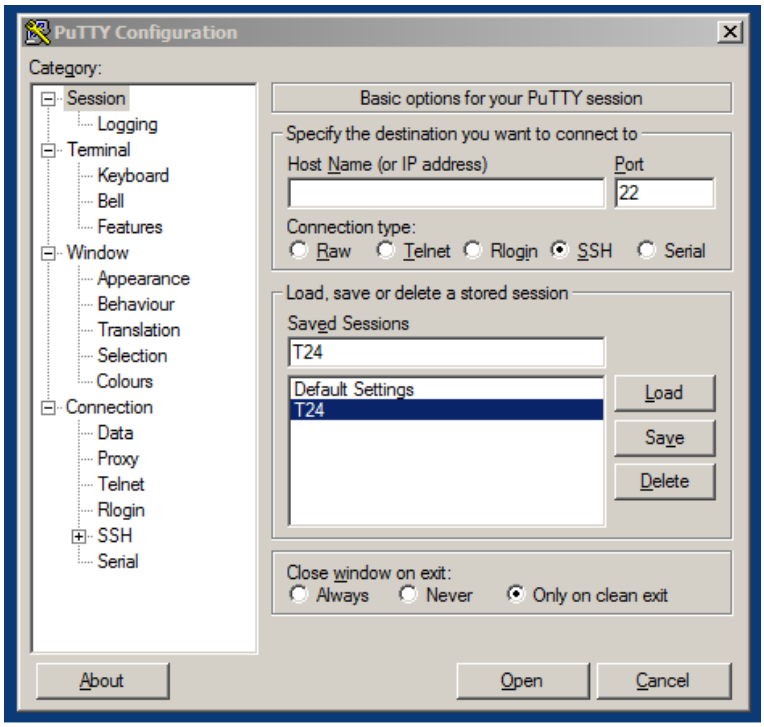


**Figure 7: Run saved session.**

*(If you need to correct some settings, select saved session, click “Load”; proceed with the changes, select session name again and click “Save”).*

Connect, supply user name and password; we’re now at jsh prompt.  Then:

jsh

```
ETS
```

Output

```
Terminal type set to EBS-JBASE
```

jsh

```
EX
```

Screen 1

```
GLOBUS Rev. R14         SIGN.ON         Copyright (c) Temenos Systems Ltd 2014


--------------------------------------------------------------------------------
     $$$$$$$$$$ $  $ $ $$   $$$$$        $$$$$    $$$ $$$$$$$$$$$$$$$$$$$
    $$$$$$$$$$$$$$$$$$ $$$  $$$   $     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
     $$$$$$$$$$$$$   $$    $$         $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      $    $$$$$$$$   $$$$           $  $$  $$$$$$$$$$$$$$$$$$$$    $$
            $$$$$$$$$$$$$$         $$ $$$$$$$$$$$$$$$$$$$$$$$$$$   $
             $$$$$$$$$$$ $           $$$$$$$$$$$$$$$$$$$$$$$$$$$
             $$$$$$$$$$            $$  $$$$  $$ $$$$$$$$$$$$$$$  $
              $$$$$$$$             $$$$     $$$$$$$$$$$$$$$$$ $ $
                $$$                $$$$$$$$$$$$$$$$$$$$$$$$$$$
                  $$$$             $$$$$$$$$$$$$    $$   $$$  $
                      $$$$$$        $$$$$$$$$$$$           $  $
                      $$$$$$$$          $$$$$$             $ $$   $$$
                       $$$$$$$          $$$$$$ $               $$$$$
                        $$$$$            $$$$  $             $$$$$$$$
                        $$$                $$                $$  $$$     $
                        $$                                             $
--------------------------------------------------------------------------------
07 OCT 2014 08:55:31  USER                              [22818,IN]
ACTION _
PLEASE ENTER YOUR SIGN ON NAME
```

After entering user name the password request appears:

Screen 2

```
GLOBUS Rev. R14         SIGN.ON         Copyright (c) Temenos Systems Ltd 2014


--------------------------------------------------------------------------------
     $$$$$$$$$$ $  $ $ $$   $$$$$        $$$$$    $$$ $$$$$$$$$$$$$$$$$$$
    $$$$$$$$$$$$$$$$$$ $$$  $$$   $     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
     $$$$$$$$$$$$$   $$    $$         $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      $    $$$$$$$$   $$$$           $  $$  $$$$$$$$$$$$$$$$$$$$    $$
            $$$$$$$$$$$$$$         $$ $$$$$$$$$$$$$$$$$$$$$$$$$   $
             $$$$$$$$$$$ $           $$$$$$$$$$$$$$$$$$$$$$$$$$$
             $$$$$$$$$$             $$  $$$$  $$ $$$$$$$$$$$$$$  $
              $$$$$$$$              $$$$     $$$$$$$$$$$$$$$$ $ $
                $$$                $$$$$$$$$$$$$$$$$$$$$$$$$$
                  $$$$             $$$$$$$$$$$$$    $$   $$$  $
                      $$$$$$        $$$$$$$$$$$$           $  $
                      $$$$$$$$         $$$$$$             $ $$   $$$
                       $$$$$$$           $$$$$$ $              $$$$$
                        $$$$$             $$$$  $            $$$$$$$$
                        $$$                 $$                $$  $$$     $
                        $$                                              $
--------------------------------------------------------------------------------
07 OCT 2014 08:55:31  USER                              [22818,IN]
ACTION _
PLEASE ENTER YOUR PASSWORD
```

If all is correct, we’re in:

Screen 3

```
Model Bank R14          SELECT APPLICATION


--------------------------------------------------------------------------------









-- LAST SIGN.ON, DATE: 05 OCT 2014     TIME: 10:17        ATTEMPTS: 0 --------
07 OCT 2014 08:54:07  USER (22 APR) INPUTTER          [15015,IN]
ACTION _
AWAITING APPLICATION
```

Go back to jsh:

T24 “AWAITING APPLICATION” prompt

```
BK
```

We can find Putty settings in the registry:

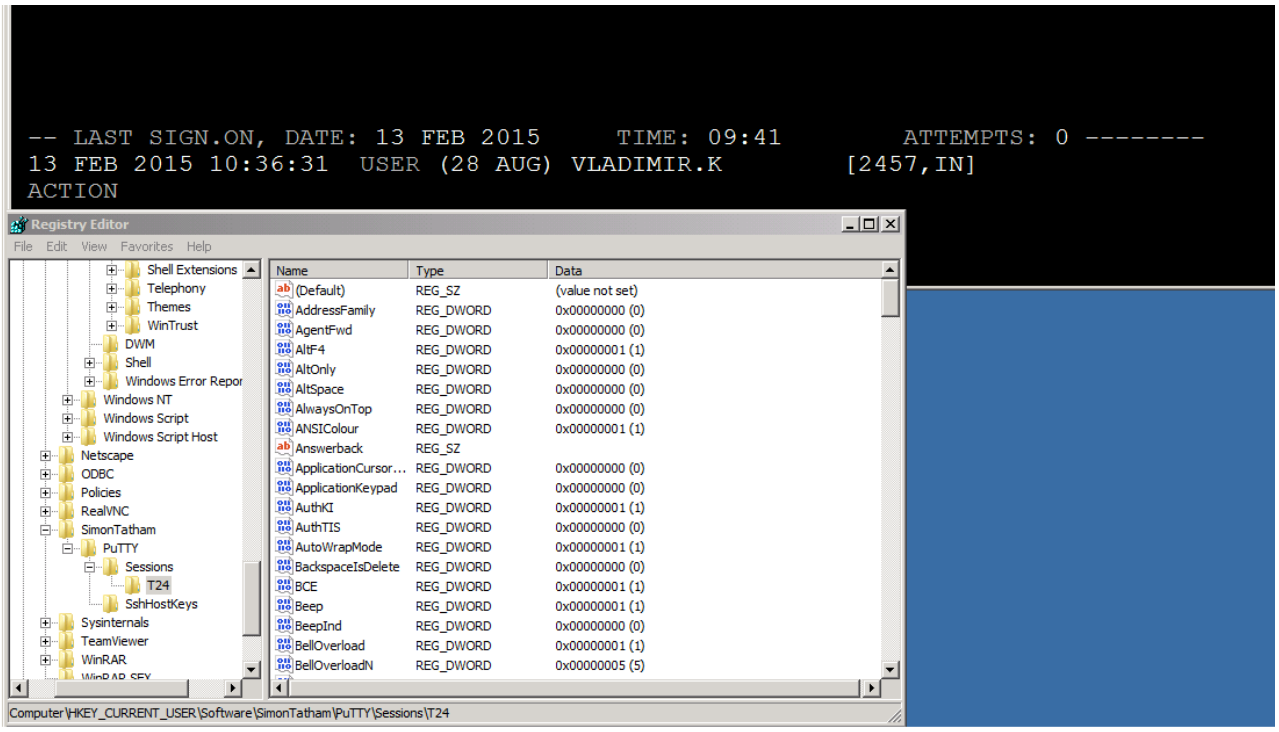


**Figure 8: Registry entry for Putty.**

Putty can be launched invoking settings directly - without settings screen; for the example above having “T24” settings:

Windows command line prompt

```
putty.exe @T24
```

*(Terminal mode T24 is often called “Classic interface”.)*

## 1.10 Logging in using Browser

To be able to work in T24 Browser we need to start the following tools:

- Web application server like JBoss (at T24 application server or at the separate server).
- jBASE agent (at T24 application server).

The latter can be done by logging in using Putty and typing at jsh prompt:

jsh

```
jbase_agent -p 20014
```

Typical output:

jbase agent session

```
3092 SocketAcceptor.h:108] starting up jAgent, Process Per Connection mode, listening
     on port 20014
```

The session should be left active (unless you start jbase agent in phantom mode using zh prefix at the above command):

jsh

```
zh jbase_agent -p 20014
```

Output

```
TASK  STARTED ON LINE 1000 FOR ACCOUNT r14 Id c25bcc24-95d2-4306-a323-4c17e6722974
```

Point your browser to:

Browser address line

```
http://127.0.0.1:9095/BrowserWeb
```

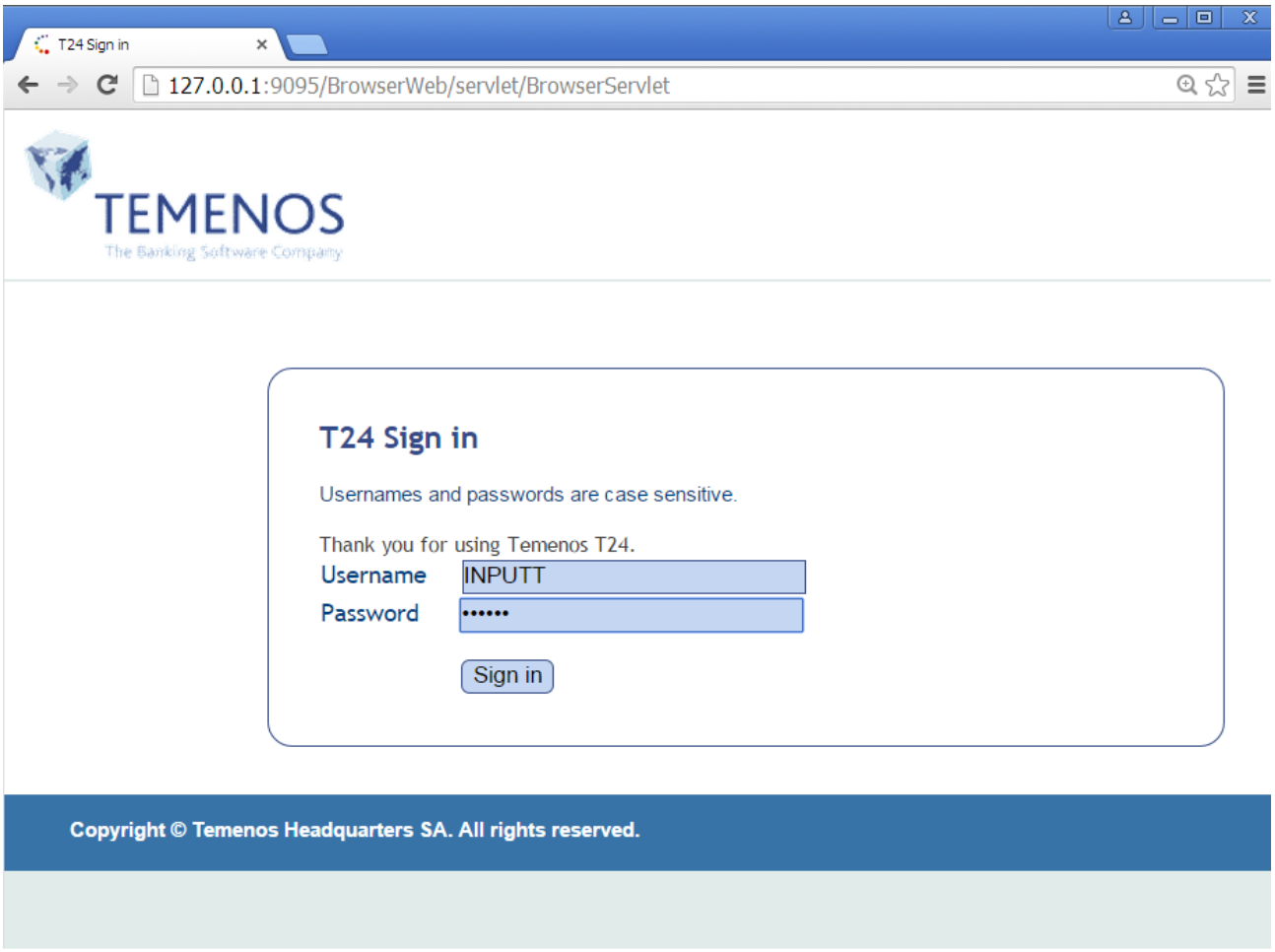


**Figure 9: T24 login screen.**

Enter T24 login name and password; you're in.

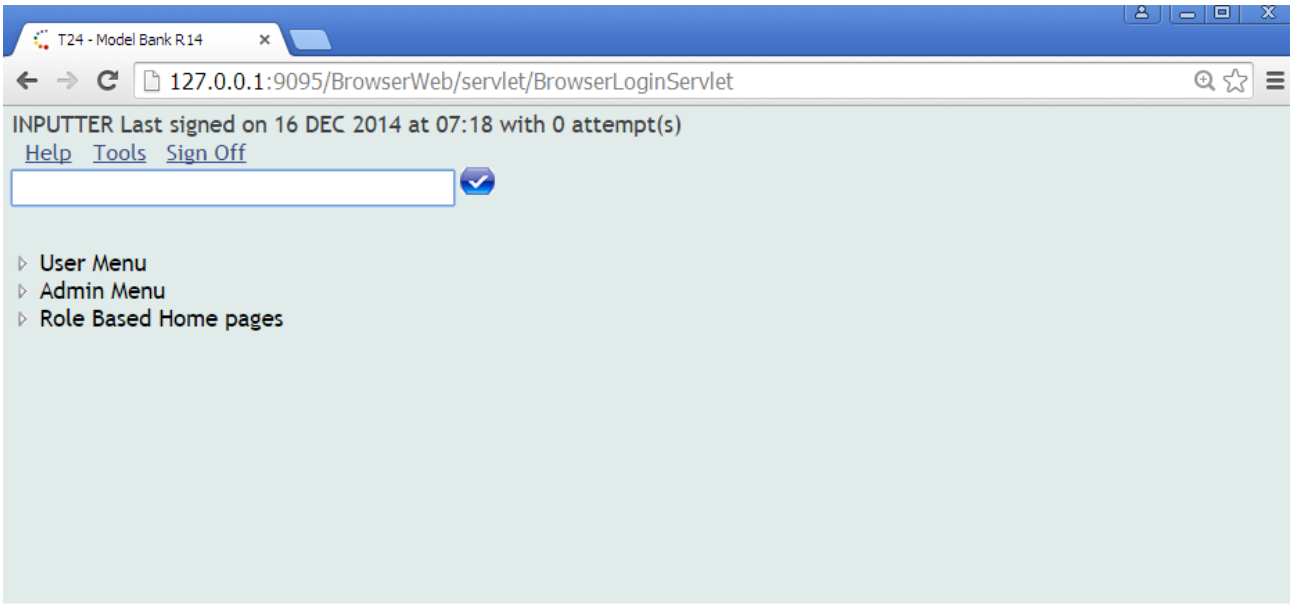


**Figure 10:  T24 - user logged in.**

If we log in using the same user @ID without logging out at another session, the newer session takes over and the older one is invalidated:

T24

```
Model Bank R14          SELECT APPLICATION


------------------------------------------------------------------------------









-- LAST SIGN.ON, DATE: 16 DEC 2014     TIME: 07:39        ATTEMPTS: 0 --------
16 DEC 2014 08:13:38  USER (22 APR) INPUTTER          [15525,IN]
ACTION AC                          Sorry, but your session is no longer active
AWAITING APPLICATION               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Or - in Browser:

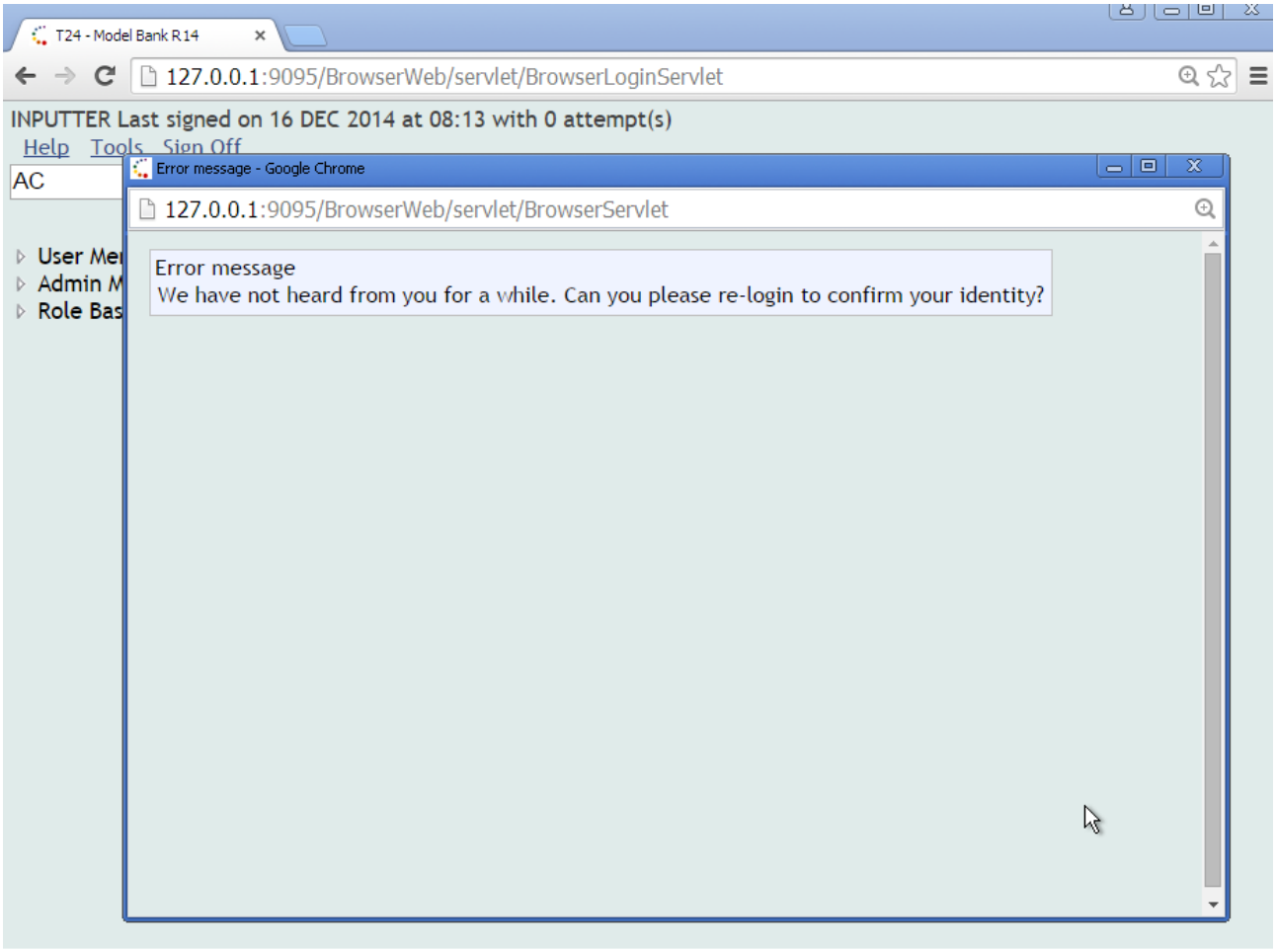


**Figure 11: T24 session invalidated.**

If something is wrong with web server (or it's not running), the error like "this webpage is not available" appears (depending of used web browser).

If web server is active - its main page should be accessible:

Browser address line

```
http://127.0.0.1:9095
```

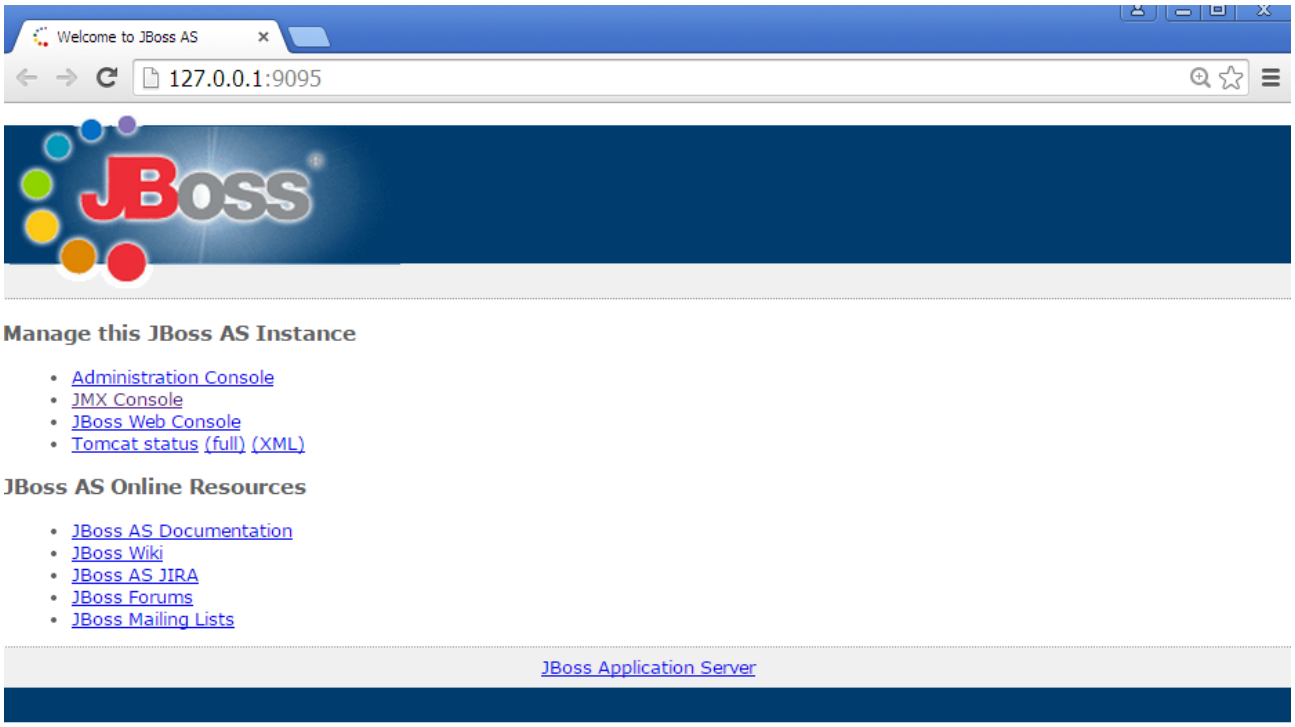


**Figure 12: JBoss main page.**

If instead of T24 login page the error appears - like one at the following screen - something is wrong with deployment of BrowserWeb.war which is supplied by Temenos.

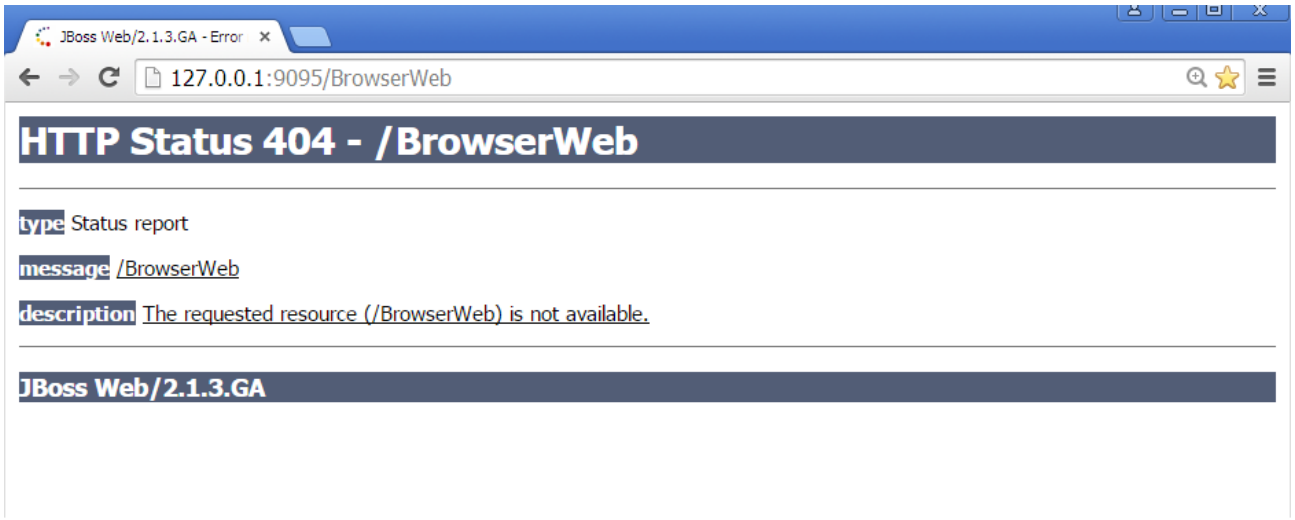


**Figure 13: Error 404.**

If login page appears but there’s an error after entering T24 user name and password - the jbase_agent is possibly not started (or its port doesn’t match the setting in t24-ds.xml:


**Figure 14: Login error.**

*Some navigation examples.*

See T24 SPF application (that we listed earlier at jsh):

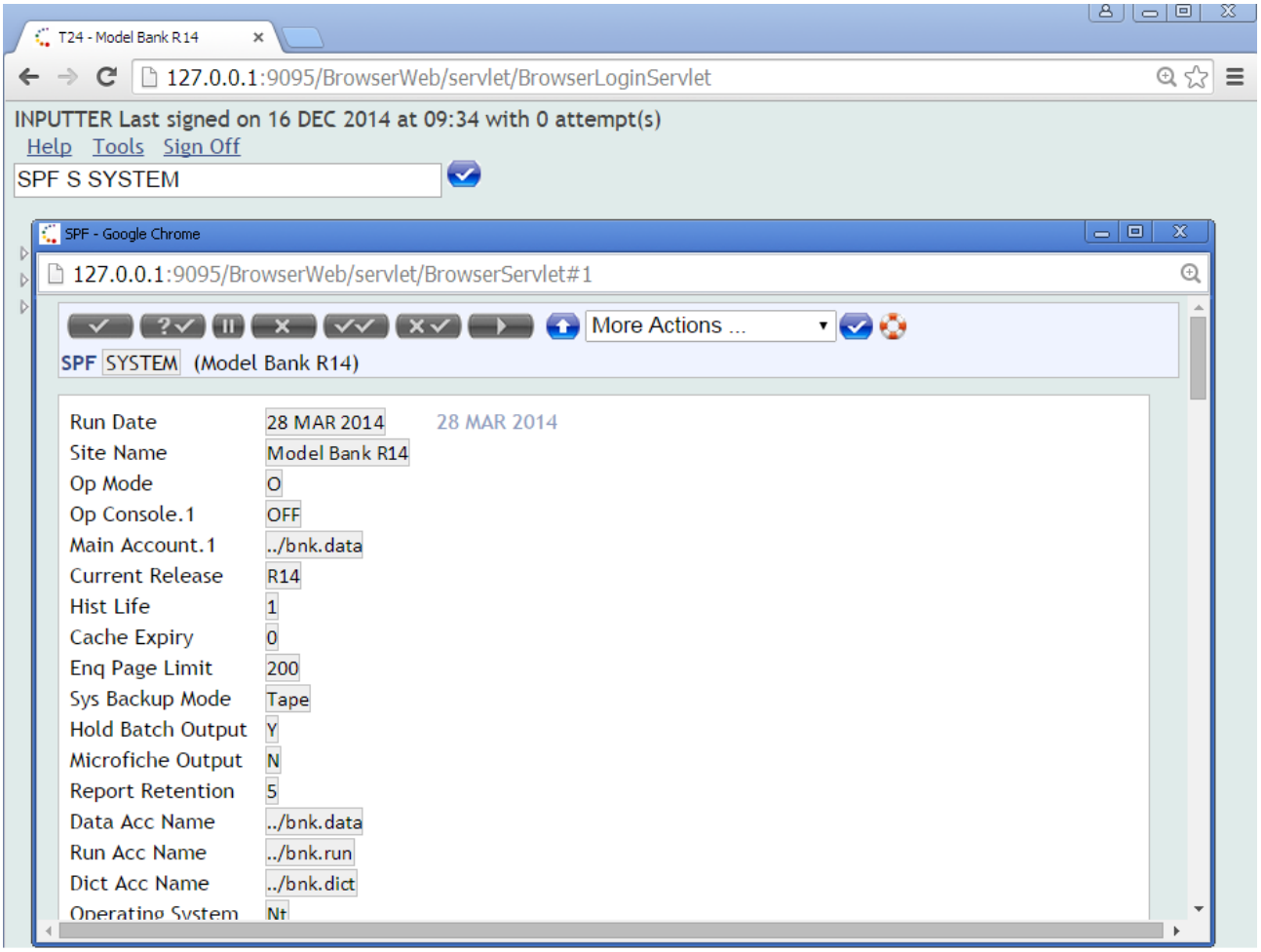


**Figure 15: See SPF record.**

See help for particular field - click on field name, e.g.  Current Release:

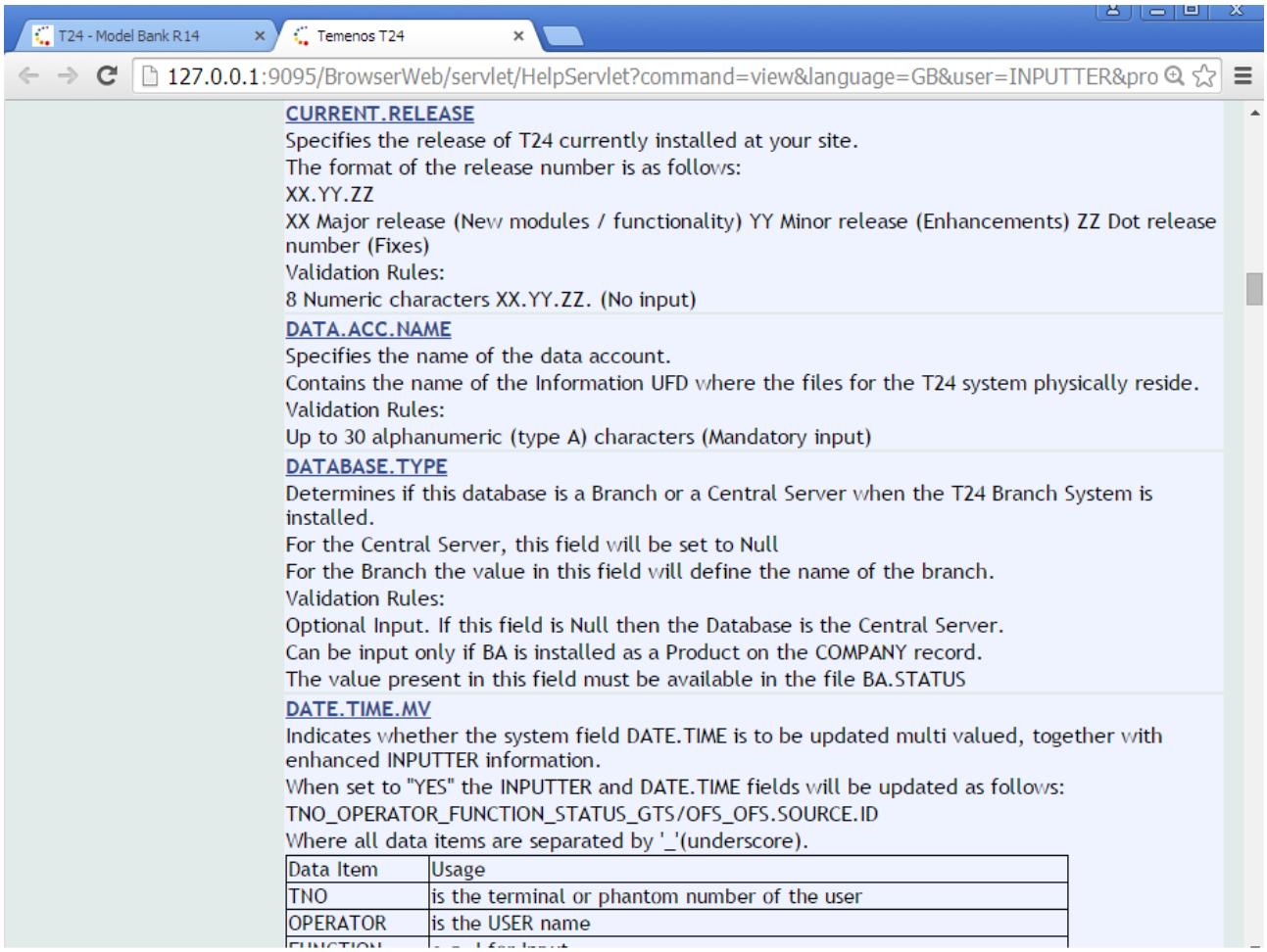


**Figure 16:  See field help.**

To log off it’s better to click “Sign Off” link at the main screen rather than close window using upper right corner “x” button.

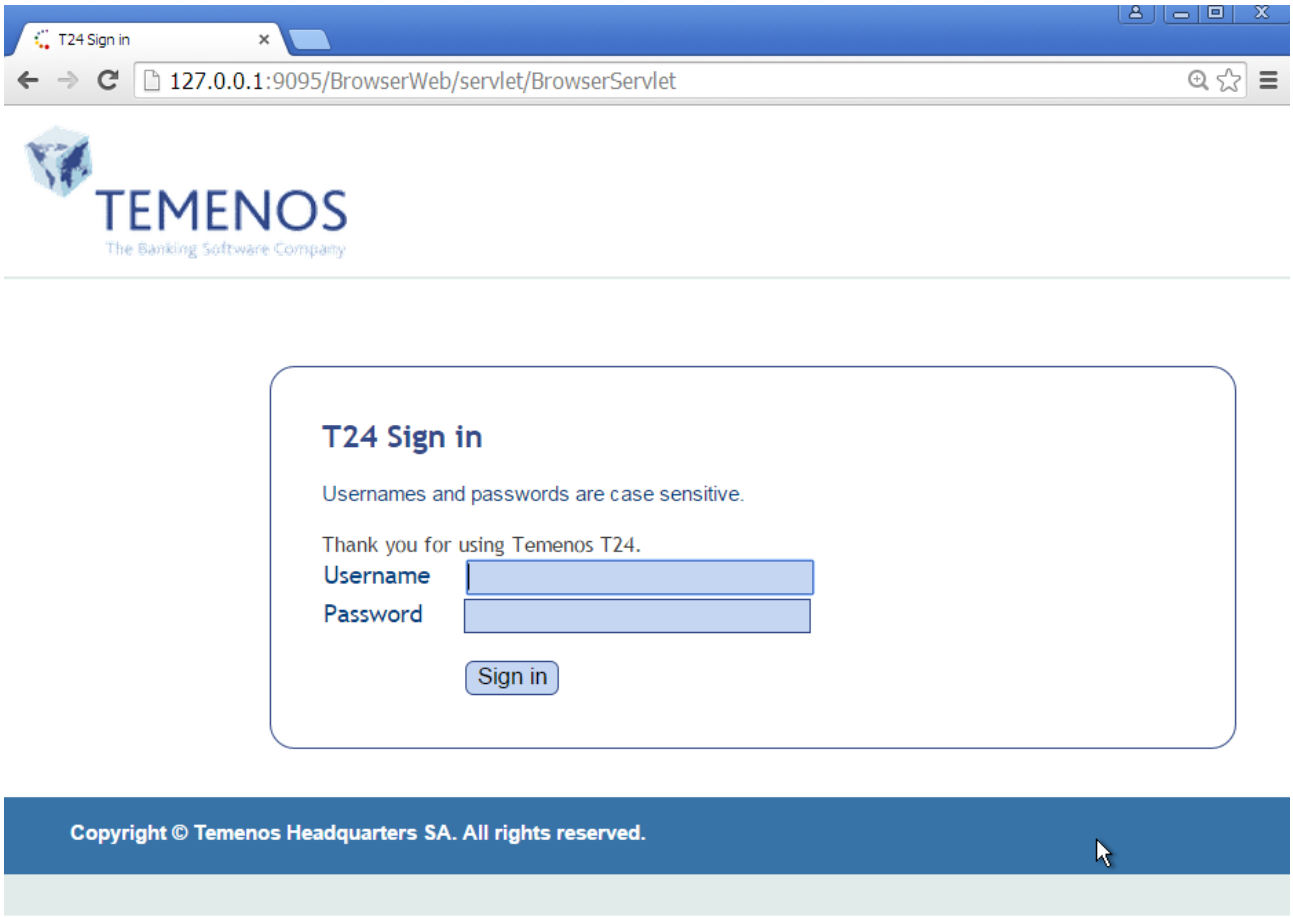


**Figure 17:  End of session.**

## 1.11  What is “T24 application” mentioned earlier?

T24 application is a combination of at least one subroutine containing table structure and business logic (based on so-called “application template”), one or several data files, dictionary file and several setup records.  See all this for application CUSTOMER.

Source code information:

jsh

```
jshow -c CUSTOMER
```

Output

```
Subroutine:D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\t24lib\st_customer\lib0.dll
                jBC CUSTOMER version 14.0 Wed May 14 16:56:45 2014
                jBC CUSTOMER source file source/r14/WIN64_TAFCR14GA
```

“Insert” file containing field names and numbers:

jsh

```
CT T24.BP I_F.CUSTOMER
```

Output

```
001 * File Layout for CUSTOMER Created 23 AUG 12 at 06:01AM by patest2
002 *      PREFIX[EB.CUS.]      SUFFIX[]
003                   EQU EB.CUS.MNEMONIC TO 1, Customer_Mnemonic TO 1,
004                     EB.CUS.SHORT.NAME TO 2, Customer_ShortName TO 2,
005                         EB.CUS.NAME.1 TO 3, Customer_Name1 TO 3,
006                         EB.CUS.NAME.2 TO 4, Customer_Name2 TO 4,
007                         EB.CUS.STREET TO 5, Customer_Street TO 5,
...
```

*(Old syntax in the left column, new one in the right.  The former starts from the prefix - EB.CUS in this case, the latter - from table name.)*

*EQU is a short form of keyword EQUATE - the method of declaring a constant in jBC. For example, Customer_ShortName will be replaced by 2 at compile time.*

List dictionary item #2:

jsh

```
CT DICT FBNK.CUSTOMER 'SHORT.NAME'
```

Output

```
    SHORT.NAME
001 D
002 2
003
004 SHORT.NAME
005 35L
006 M
007
...
```

*(“M” in field 6 means “Multi-valued field”.)*

Dictionary in T24 is built upon authorisation of corresponding STANDARD.SELECTION record.  (So - never add dictionary entries to T24 tables manually since they might be overwritten.)  To see the field SHORT.NAME type SS S CU at T24 “AWAITING APPLICATION” prompt and navigate to page 3 using F3 key to list pages down one by one or entering P3 command.

T24

```
Model Bank R14          STANDARD SELECTION FIELDS SEE

    FILE.NAME......... CUSTOMER
--------------------------------------------------------------------------------
 11. 4 SYS.LANG.FIELD. N
 12. 4 SYS.GENERATED.. Y
  1. 5 SYS.FIELD.NAME. SHORT.NAME                          <===== from here
  2. 5 SYS.TYPE....... D
  3. 5. 1 SYS.FIELD.NO 2
  4. 5. 1 SYS.VAL.PROG IN2SWI
  6. 5 SYS.DISPLAY.FMT 35L
  7. 5 SYS.ALT.INDEX.. N
 10. 5 SYS.SINGLE.MULT M
 11. 5 SYS.LANG.FIELD. Y
 12. 5 SYS.GENERATED.. Y                                   <===== up to here
  1. 6 SYS.FIELD.NAME. NAME.1
  2. 6 SYS.TYPE....... D
  3. 6. 1 SYS.FIELD.NO 3
  4. 6. 1 SYS.VAL.PROG IN2SWI
  6. 6 SYS.DISPLAY.FMT 35L
--------------------------------------------------------------------------------
07 OCT 2014 11:35:28  USER (22 APR) INPUTTER          [1748,INPAGE 3   >>116>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

SS and CU used in this command are both shortcuts that save typing.  You can see them in T24 - application ABBREVIATION - or at jsh level.  Go back to “AWAITING APPLICATION” prompt using F1, then - back to jsh (BK or PGM.BREAK), then:

jsh

```
LIST F.ABBREVIATION 'SS' 'CU'
```

Output

```
ABBREVIATION.CODE. SS
ORIGINAL.TEXT..... STANDARD.SELECTION
RECORD.STATUS.....
CURR.NO.......... 1
INPUTTER......... 93958_OFFICER__OFS_SEAT
DATE.TIME........ 1404081706
AUTHORISER....... 93958_OFFICER_OFS_SEAT
CO.CODE.......... GB0010001
DEPT.CODE........ 1
AUDITOR.CODE......
AUDIT.DATE.TIME...

@ID.............. CU
ABBREVIATION.CODE. CU
ORIGINAL.TEXT..... CUSTOMER
RECORD.STATUS.....
CURR.NO.......... 1
INPUTTER......... 44113_OFFICER__OFS_SEAT
DATE.TIME........ 1404081706
AUTHORISER....... 44113_OFFICER_OFS_SEAT
```

```
CO.CODE.......... GB0010001
DEPT.CODE........ 1
AUDITOR.CODE......
AUDIT.DATE.TIME...
```

Back to T24 application definition.  PGM.FILE is another setup table:

jsh

```
LIST F.PGM.FILE 'CUSTOMER' TYPE
```

*(Again, in quotes is @ID, without quotes - dictionary entry; a data field TYPE in this case.)*

Output

```
@ID.................................    TYPE

CUSTOMER                                H
```

*(Meaning that application has history)*

Another setup table:

jsh

```
LIST F.FILE.CONTROL 'CUSTOMER'
```

Output

```
@ID.............. CUSTOMER
DESCRIPTION...... Customer details
PRODUCT.......... ST
SUFFIXES......... $HIS   $NAU   $DEL
FILE.TYPE........ 2
FILE.MODULO...... 31
CLASSIFICATION... CUS
...
```

*(“CUS” means that files prefix contains company mnemonic.  It was not the case for ABBREVIATION that has type INT, i.e.  one file for all branches.)*

See VOC (multi-purpose system file) entry for CUSTOMER file, assuming company mnemonic:

jsh

```
CT VOC FBNK.CUSTOMER
```

Output

```
        FBNK.CUSTOMER
001 F
002 ../bnk.data/st/FBNK_CUSTOMER
003 ../bnk.dict/F_CUSTOMER]D
```

Finally, list contents of all data files (one record for each of them).

“Live”:

jsh

```
LIST FBNK.CUSTOMER SAMPLE 1
```

Output

```
@ID.......................... 188888
CUSTOMER.CODE................ 188888
MNEMONIC..................... BBHINCUS
SHORT.NAME................... BBH Inhouse Custodian
NAME.1....................... BBH Inhouse Custodian
NAME.2....................... BBH Inhouse Custodian
STREET....................... 125 Finsbury Pavement
TOWN.COUNTRY................. LONDON
...
```

“Unauthorised”:

jsh

```
LIST FBNK.CUSTOMER$NAU SAMPLE 1
```

Output

```
@ID.......................... 100226
CUSTOMER.CODE................ 100226
MNEMONIC..................... COOPERA
SHORT.NAME................... Coopera
NAME.1....................... Andrew Cooper
NAME.2....................... Andrew Cooper
STREET....................... 20 Craven Park
TOWN.COUNTRY................. Greater London
...
```

“History”:

jsh

```
LIST FBNK.CUSTOMER$HIS SAMPLE 1
```

Output

```
@ID.......................... 100336;2
CUSTOMER.CODE................ 100336;2
MNEMONIC..................... GERLING
SHORT.NAME................... Rolf Gerling
NAME.1....................... Rolf Gerling
NAME.2.......................
STREET....................... Haupt Street 18
TOWN.COUNTRY................. Musterstadt
...
```

“Deleted” (recently introduced type):

jsh

```
LIST FBNK.CUSTOMER$DEL SAMPLE 1
```

Output

```
@ID.......................... 150501;14051624262246
CUSTOMER.CODE................ 150501;14051624262246
MNEMONIC..................... HENAO
SHORT.NAME................... COSMIN
NAME.1....................... MOLDOVAN
NAME.2.......................
STREET....................... Robinson Road
TOWN.COUNTRY................. SINGAPORE
...
```

See fields CURR.NO and RECORD.STATUS that are part of record’s “audit trail”:

jsh

```
LIST FBNK.CUSTOMER CURR.NO RECORD.STATUS SAMPLE 1
```

Output

```
@ID.......    CURR.NO    RECORD.STATUS

    188888          1
```

jsh

```
LIST FBNK.CUSTOMER$NAU CURR.NO RECORD.STATUS SAMPLE 1
```

Output

```
@ID.......    CURR.NO    RECORD.STATUS

    100226          3    INAU
```

jsh

```
LIST FBNK.CUSTOMER$HIS CURR.NO RECORD.STATUS SAMPLE 1
```

Output

```
@ID.......    CURR.NO    RECORD.STATUS

  100336;2          2
```

jsh

```
LIST FBNK.CUSTOMER$DEL CURR.NO RECORD.STATUS SAMPLE 1
```

Output

```
@ID.......    CURR.NO    RECORD.STATUS

         1          1    INAU
50501;1405
1624262246
```

See data file statistics:

jsh

```
jstat -v FBNK.CUSTOMER
```

Output

```
File Path       = ..\bnk.data\st\FBNK_CUSTOMER
File Type       = JR,           Hash method   = 5, Created = Mon May 19 21:52:52 2008
Frame size      = 4096,         OOG Threshold   = 2048
File size       = 909312,       Freespace       = 0 frames
Internal Modulo = 3/7/19,       External Modulo = 31
Inode no.       = 91644,        Device no.      = 30017
Accessed        = Sun Oct 05 08:57:08 2014,   Modified     = Sun Oct 05 08:57:08 2014
Backup  = YES, Log      = YES,  Rollback        = YES,  Secure updates  = YES
Deallocate pointers : NO        Deallocate frames NO
Revision level  = 1
Record Bytes    = 177788,       Record Count    = 392
Bytes/Record    = 453,  Bytes/Group     = 804
Data Frames     = 221,  Ptr Frames      = 0
OOG Bytes       = 0,    OOG Frames      = 0
Sum Squares     = 82498122,     Std Dev Mean    = 525
```

## 1.12 Enter the application at T24 “AWAITING APPLICATION” prompt

Applications are case-sensitive so if we type “customer” or “Customer” we get the error:

T24

```
Model Bank R14          SELECT APPLICATION


------------------------------------------------------------------------------









-- LAST SIGN.ON, DATE: 15 DEC 2014     TIME: 06:05        ATTEMPTS: 0 --------
15 DEC 2014 07:00:35  USER (22 APR) INPUTTER          [10397,IN]
ACTION customer                                        APPLICATION MISSING
AWAITING APPLICATION                                   ^^^^^^^^^^^^^^^^^^^
```

Correct input is in upper case - CUSTOMER (or CU which is the abbreviation for CUSTOMER):

Screen 1

```
Model Bank R14          SELECT APPLICATION


------------------------------------------------------------------------------
...




-- LAST SIGN.ON, DATE: 07 OCT 2014     TIME: 09:04        ATTEMPTS: 0 --------
07 OCT 2014 09:33:17  USER (22 APR) INPUTTER          [3065,IN]
ACTION CUSTOMER_
AWAITING APPLICATION
```

Header changed; we’re now at “AWAITING FUNCION” prompt:

Screen 2

```
Model Bank R14          CUSTOMER


--------------------------------------------------------------------------------









-- LAST SIGN.ON, DATE: 07 OCT 2014     TIME: 09:04        ATTEMPTS: 0 --------
07 OCT 2014 09:33:17  USER (22 APR) INPUTTER          [3065,IN]
ACTION _
AWAITING FUNCTION
```

Function L - list live file:

T24

```
Model Bank R14

CUSTOMER - list of live file
--------------------------------------------------------------------------------
1  111615 ABCE       ABC EUROPE                      Implementation        U|  G|
2  111612 ABCG       ABC GLOBAL                      Implementation        U|  U|
3  111616 ABCL       ABC UK                          Implementation        U|  U|
4  111617 ABCS       ABC SINGAPORE                   Implementation        U|  S|
5  111614 ABCT       ABC TRUST                       Implementation        U|  U|
6  111613 ABCU       ABC USA                         Implementation        U|  U|
7  100112 ABNAMRO    Abn Amro Securities             Chief Securities De|  N|  N|
8  100369 ABUDHAOIL  Abu Dhabi National Oil Company  Corporate Loan Supe|  U|  U|
9  128046 ADMIN      Administrator Of Acorn Engg CompMortgage Dept User 2  U|  U|
10  100396 AIRBOURNE  Airbourne Freight               Customer Service Ag|  R|  U|
11  128072 AIRBUS     Airbus                          Trade Finance Offic|  G|  G|
12  129009 AKAI       AKAI                            Trade Finance Offic|  J|  J|
13  130130 ALBERT     ALBERT                          Implementation        G|  G|
14  128002 ALEX       Alex                            Branch Operations M|  U|  U|
15  111424 ALEXME     Alex Robert                     Retail Banking Mana|  U|  U|
16  129018 ALEXP      ALEX                            Implementation        G|  G|
17--100301-ALLEGRA   -Allegra                        Corporate Officer   --U|- U|
187 1009711ALLIANCE   Alliance Investment Managers    Chief SecPAGE   1 >>>3>>>
19CT100433 AMEXINDONE American Expr
9AWAITING PAGE INSTRUCTIONS
```

*Page down - F3 (or Ctrl-F, Enter if your terminal emulator isn't set up to handle T24 keys), page up - F2 (or Ctrl-B, Enter), one level up - F1 (or Ctrl-U, Enter), see record in position 2 - 2 Enter S Enter F5 (or Ctrl-V Enter instead of F5), edit record in position 10 - 10 Enter I Enter F5.*

## 1.13 Edit a record

T24

```
Model Bank R14          CUSTOMER INPUT

    CUSTOMER.CODE..... 100396
--------------------------------------------------------------------------------
  1 MNEMONIC.......... AIRBOURNE
  2. 1 GB SHORT.NAME.. Airbourne Freight
  3. 1 GB NAME.1...... Airbourne Freight
  4. 1 GB NAME.2......
  5. 1 GB STREET...... No. 415
  6. 1. 1 GB ADDRESS..
  7. 1 GB TOWN.COUNTRY Hsin Yi Road Section 4
  8. 1 GB POST.CODE...
  9. 1 GB COUNTRY..... Taipei
 10. 1 RELATION.CODE..
 11. 1 REL.CUSTOMER...
 12. 1 REVERS.REL.CODE
 13. 1. 1 REL.DELIV.OP
 14. 1. 1 ROLE........
 15. 1. 1 ROLE.MORE.IN
 16. 1. 1 ROLE.NOTES..
--------------------------------------------------------------------------------
07 OCT 2014 09:43:11  USER (22 APR) INPUTTER          [3065,INPAGE 1   >>12>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

*Page down - F3, page up - F2, end - F4 (or Ctrl-E Enter), up to page 1 - P1 Enter, go to field GB NAME.2 - 4.1 Enter, expand it - < Enter, delete expanded value - - Enter, navigate between fields - F3/F2, got to "ACTION" line - F4, return - F1.*

We're now at "AWAITING ID" prompt.  E Enter - list of NAU file, F1 - back, F1 - back to "AWAITING APPLICATION" prompt.

## 1.14 T24 functions

In the previous chapter we used "S" in keys sequence to see a record and "I" - to edit.  These letters represent 2 T24 functions.

The full list is:

- I - input.
- S - see.
- L - list live file.
- E - list NAU file.

- P - print.
- D - delete (NAU record).
- R - reverse (authorised record).
- H - history restore.
- C - copy.
- A - authorise input or reversal.
- V - verify.
- Q - audit.

As we know all necessary functions now, we can create personal T24 user by copying an existing one:

T24 “AWAITING APPLICATION” prompt

```
USER, S INPUTTER
F1
C
```

Supply @ID, e.g.  TRAIN.01 at “AWAITING ID” prompt:

T24

```
Model Bank R14          USER PROFILE,  COPY

    USER.ID........... TRAIN.01
--------------------------------------------------------------------------------
  1 USER.NAME......... INPUTTER
  2 SIGN.ON.NAME......
  3 CLASSIFICATION.... INT
  4 LANGUAGE.......... 1                   English
  5. 1 COMPANY.CODE... GB0010001           Model Bank R14
  5. 2 COMPANY.CODE... GB0010002           MF LEAD COMPANY
  5. 3 COMPANY.CODE... EU0010001           Model Bank - Europe
  5. 4 COMPANY.CODE... SG0010001           Model Bank - Singapore
  5. 5 COMPANY.CODE... GB0010005           Model Bank Branch 1
  5. 6 COMPANY.CODE... GB0010003           MF Branch 1
  5. 7 COMPANY.CODE... GB0010004           MF Branch 2
  6 DEPARTMENT.CODE... 1                   Implementation
  7 PASSWORD.VALIDITY. 01 NOV 2014 M0601   01 NOV 2014 Every 6 months on day 1
  8 START.DATE.PROFILE 15 MAY 2014
  9 END.DATE.PROFILE.. 20 SEP 2016
 10. 1 START.TIME..... 00:00
--------------------------------------------------------------------------------
07 OCT 2014 13:10:40  USER (22 APR) INPUTTER          [14805, PAGE 1   >>>6>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

Amend fields 1, 2 and 8 (use system date - not bank date - for START.DATE.PROFILE field):

T24

```
Model Bank R14          USER PROFILE,  COPY

    USER.ID........... TRAIN.01
------------------------------------------------------------------------------
  1 USER.NAME......... TRAINEE 01
  2 SIGN.ON.NAME...... TRAIN01
  3 CLASSIFICATION.... INT
  4 LANGUAGE.......... 1                  English
  5. 1 COMPANY.CODE... GB0010001          Model Bank R14
  5. 2 COMPANY.CODE... GB0010002          MF LEAD COMPANY
  5. 3 COMPANY.CODE... EU0010001          Model Bank - Europe
  5. 4 COMPANY.CODE... SG0010001          Model Bank - Singapore
  5. 5 COMPANY.CODE... GB0010005          Model Bank Branch 1
  5. 6 COMPANY.CODE... GB0010003          MF Branch 1
  5. 7 COMPANY.CODE... GB0010004          MF Branch 2
  6 DEPARTMENT.CODE... 1                  Implementation
  7 PASSWORD.VALIDITY. 01 NOV 2014 M0601  01 NOV 2014 Every 6 months on day 1
  8 START.DATE.PROFILE 10 OCT 2014
  9 END.DATE.PROFILE.. 20 SEP 2016
 10. 1 START.TIME..... 00:00
------------------------------------------------------------------------------
07 OCT 2014 13:10:40  USER (22 APR) INPUTTER          [14805, PAGE 1   >>>6>>>
ACTION
```

Commit (F5).  Go back to jsh, log in using new SIGN.ON.NAME (not @ID!), supply new password not shorter than 6 characters, repeat it at the next prompt; now we're in:

T24 - logged in as a new user

```
-- LAST SIGN.ON, DATE: ---             TIME: ---          ATTEMPTS: 5 --------
02 FEB 2015 14:38:58  USER (22 APR) TRAIN.01          [30653,IN]
ACTION
```

*(It's not possible however to type USER, C INPUTTER at T24 "AWAITING APPLICATION" prompt - we get the error "NO COPY AVAILABLE". The method described above - USER S ...  F1 ...  C is a workaround.)*

## 1.15  See who's in the system.

jsh

```
WHERE
```

Output

```
 Port   Device     Account      PID      Command
 1      VT100      t24          2308     ***** Thread type Normal
*2      VT100      r14          2484     jsh
                                         WHERE
 21     VT100      t24          624      ***** Thread type Normal
```

Full detail for all or for particular port:

jsh

```
WHERE (V 21
```

Output

```
 Port   Device     Account      PID      Command
 21     VT100      t24          624      Port 21 details at 11:37:50 05 OCT 2014
                                         Thread type Normal
                                         Usage Count 48 from 100K
                                         Application Open Files 1 ,
                                         Actual O/S Open Files 1
                                         Memory: Free 0   Used 2,334,720
                                         READ's 0 , WRITE's 0
                                         DELETE's 0 , CLEARFILE's 0
                                         EXECUTE's 0 , INPUT's 0
                                         OPEN's 0
```

Constant monitoring:

jsh

```
MW42 -a
```

Sample output

```
jBASE 14.0 Monitor (2.5) - WinNT                          10:03:35  07 OCT 2014

Port        User     Pid      Files Perf  Del  Read Write Open   Mem   Cpu  Prog
    1               2308      1 (1)    0    0     0     0    0 10.5M
*   2 TRAIN.01-0    2484    93 (82)   75   25 10362    97  738 23.9M  0.00  MW42 -a
                                                                            (mw42.b,708)
   21                624      1 (1)    0    0     0     0    0 2.23M
```

*(Exit - Q)*

*(T24 user ID - TRAIN.01 - is still retained for the port #2 even after logout.)*

Show time at the column 1 (useful when MW42 output is being logged via COMO ON or into putty.log):

jsh

```
MW42 -a -t
```

Sample output

```
jBASE 14.0 Monitor (2.5) - WinNT                          05:25:17  25 FEB 2015

      Port  User  Pid Files Perf Del Read Write Open Mem Cpu Prog
05:25  1           2056   1 (1) 0 0   0  0   0 10.5M
05:25 *2           1872   7 (6) 2 1 137  3   8 13.3M    3 MW42 -a -t (mw42.b,708)
05:25  3 INPUTTER- 2052 82 (75) 1 4 358 22 106 23.9M 28.0 jbase_agent (jbase_agent,424
05:25  4           2484 40 (39) 1 1 165  3  63 18.7M  0.0 jbase_agent (jbase_agent,72)
05:25  5           2696 40 (39) 1 1 169  3  64 18.6M  0.0 jbase_agent (jbase_agent,72)
05:25  6           2744 40 (39) 1 1 173  3  65 18.6M  0.0 jbase_agent (jbase_agent,72)
05:25  7           2044 40 (39) 1 1 177  3  66 18.7M  0.0 jbase_agent (jbase_agent,72)
05:25  8           2756 40 (39) 1 1 181  3  67 18.7M  0.0 jbase_agent (jbase_agent,72)
```

## 1.16 First jBC program

A program may reside either in bnk.run (like short test ones) or in “aaa.BP” directory, where “aaa” is an appropriate name, e.g. LOCAL.BP, MODEL.BP etc. Create such directory:

jsh

```
CREATE-FILE DATA ETC.BP TYPE=UD
```

Output

```
[ 417 ] File ETC.BP created , type = UD
```

*(We don’t need a dictionary that would become just another folder “ETC.B]D” hence the keyword DATA.)*

Of course this directory could also be created using OS command.

Choose program name; see whether it’s already used:

jsh

```
jshow -c PROG.0
```

If there’s no output, name can be used. Launch jBASE editor (or use a text editor to create this file in ETC.BP):

jsh

```
JED ETC.BP PROG.0
```

In JED window type:

CRT ’It takes’, RND(1000) + 180, ’days to start understanding T24’
Press Enter
Press Esc
Type FI
Press Enter

See the source code again:

jsh

```
CT ETC.BP PROG.0
```

Output

```
    PROG.0
001 CRT 'It takes', RND(1000) + 180, 'days to start understanding T24'
002
```

Compile:

jsh

```
BASIC ETC.BP PROG.0
CATALOG ETC.BP PROG.0
```

Output

```
PROG.0
BASIC_2.c
Source file PROG.0 compiled successfully
PROG.0
Object PROG.0 cataloged successfully
```

See program information again:

jsh

```
jshow -c PROG.0
```

Output

```
Executable:          D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\bin\PROG.0.dll
                     jBC main() version 14.0 Tue Oct 07 13:20:08 2014
                     jBC main() source file ETC.BP
Executable (DUP!!):  D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\bin\PROG.0.exe
                     jBC main() version 14.0 Tue Oct 07 13:20:08 2014
                     jBC main() source file ETC.BP
```

*(Never mind “DUP!!”  since we’re creating both executable and shared library.)*

List ETC.BP after compilation:

jsh

```
LIST ETC.BP
```

Output

```
$PROG.0
PROG.0
```

$PROG.0 is the object module that isn’t necessary after compilation.

See where the executables are:

jsh

```
LIST bin
```

Output

```
...
PROG.0.dll
PROG.0.exe
PROG.0.ilk
PROG.0.pdb
...
```

They were created in “bin” since we have JBCDEV_BIN set accordingly:

jsh

```
echo %JBCDEV_BIN%
```

Output

```
D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\bin
```

“bin” also has to be in the PATH:

jsh

```
jdiag -v
```

Output

```
...
Program dir (Default)       : 'D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\bin'
Program Path 'D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\bin' is in your PATH
...
```

Finally, run:

jsh

```
PROG.0
```

*(Run it every time when you think you know all about T24.)*

## 1.17 Set up JED for some useful key shortcuts

Try to press keys Home, End, PgUp, PgDn, Ins in JED. See the effect:

JED

```
File ETC.BP , Record 'PROG.0'                                Insert      23:05:43
Command-> [6^
0001 CRT 'Right Ho, T24!'[4^[4^[5^
-------------------------------- End Of Record --------------------------------
```

*(To abandon this mess use JED command EX instead of FI to exit without saving.)*

To be able to use these keys, create the file .jedrc in bnk.run:

.jedrc contents

```
# comment
# make END key do EOL
set end_line = \033:\133:\064:\176
# uncouple CTRL Y from its standard setting
set noback_tab
# set CTRL Y to 'delete line'
set delete_line = \031
# make HOME key do Start of Line
set start_line = \033:\133:\061:\176
# PgUp/PgDn
set scroll_up_page = \033:\133:\065:\176
set scroll_down_page = \033:\133:\066:\176
# Ins - toggle insert/overwrite
set toggle_insert = \033:\133:\062:\176
```

That’s it.  Keys Home and End will also work in jsh.

## 1.18 What is COMO?

Stands for “command output”. A way to log the terminal output.

jsh

```
COMO ON QWERTY1
WHERE
COMO OFF
COMO ON QWERTY2
WHERE (VN
COMO OFF
```

jsh

```
CT &COMO& QWERTY1
```

Output

```
    QWERTY1
001 COMO QWERTY1 established 10:41:59  07 OCT 2014
002 jsh r14  -->.[K.[JW.[K..[CH.[K..[CE.[K..[CR.[K..[CE.[K..[C
003 .[H.[J Port   Device     Account      PID      Command

004  1      VT100      t24          2308     ***** Thread type Normal
005 *2      VT100      r14          2252     jsh
006                                          WHERE
007  21     VT100      t24          624      ***** Thread type Normal
008 jsh r14  -->.[K.[JC.[K..[CO.[K..[CM.[K..[CO.[K..[C .[K..[CO.[K..[CF.[K..[
    CF.[K..[C
009 COMO completed 10:42:06  07 OCT 2014
```

*(Trash in the output is terminal escape sequences.)*

Search the COMO directory for contents:

jsh

```
SEARCH &COMO&
String :WHERE
String :
Record Keys : *
```

Output

```
 2 Records selected

>
```

jsh

```
LIST ONLY &COMO&
```

Output

```
@ID

QWERTY1
QWERTY2

 2 Records Listed
```

## 1.19 Get more T24 tables

Now we list the system file VOC for “VOC entries” (actually - records) where contents of the first field start with “F”. If we don’t know the field name, special notation can be used:

jsh

```
LIST VOC WITH *A1 LIKE "'F'..."
             ^^^
```

Output

```
NAME.....................................    TYPE

F.DX.REVAL.DET.ENHANCED                      F
F.GENERATE.ENTRY.WRK                         F
F.GT.SYSTEM.PARAMETER$NAU                    F
F.RUN.GRP.PERF.TAKEON                        F
F.SA.RATIOS                                  F
F.SC.CCY.RANK                                F
F.SETTLEMENT.FRA                             F
FBNK.ADHOC.STATEMENTS                        F
FBNK.AM.SIZING$NAU                           F
FBNK.POSITION.TRANSFER$HIS                   F
FBNK.TABLE.CAPITALIS.CORR                    F
...
```

LIKE "’F’..." means “starting from character F”. We don’t use full comparison because there could be a comment after “F”. The length of a field can be measured using “EVAL” keyword to illustrate this:

jsh

```
LIST VOC WITH *A1 LIKE "'F'..." *A1 EVAL "LEN(F1)"
```

Output

```
VOC.........    *A1...........    LEN(F1)........

F.DX.REVAL.D    F                 1
ET.ENHANCED
F.GENERATE.E    F                 2
NTRY.WRK
F.GT.SYSTEM.    F GTS SYSTEM P    23
PARAMETER$NA    ARAMETERS
U
F.RUN.GRP.PE    F                 1
RF.TAKEON
F.SA.RATIOS     F Records fina    66
                ncial ratios a
                nd formulas fo
                r calculation
                of ratios.
...
```

LEN(*A1) doesn’t work so another field alias is used - F1.  There are others - F2, F3 but not many since they are simply VOC entries with type “D”. To see such record fully we use CT command:

jsh

```
CT VOC 'F10'
```

Output

```
    F10
001 D
002 10
003
004
005 15T
006 S
```

Generally, VOC is a multi-purpose jBASE-level system file that is used, besides what we’ve just seen, to store paragraphs (automated scripts), aliases for commands etc.  Many VOC records are obsolete but it’s a good idea not to amend or delete existing records anyway.

If we know part of the name, we can narrow the search:

jsh

```
LIST VOC WITH *A1 LIKE "'F'..." AND @ID LIKE "'F.CO'5A"
```

Here we’re trying to look for record @IDs which start from “F.CO” followed by 5 alpha characters:

Output

```
NAME.....................................    TYPE

F.COUNTRY                                    F
F.COMPANY                                    F
...
```

For XML data storage - e.g.  MSSQL - VOC entries for data files look like:

jsh

```
CT VOC FBNK.ACCOUNT
```

Output

```
    FBNK.ACCOUNT
001 TABLE
002 XMLMSSQLInit
003 D_F_ACCOUNT NOXMLSCHEMA
004 FBNK_ACCOUNT
```

## 1.20  SELECT COMPANY file - list of bank branches

jsh

```
SELECT F.COMPANY
```

Output

```
 7 Records selected

>
```

Prompt has changed - we now have an active SELECT list.  In case we don’t need it - use the following command:

jsh

```
CLEARSELECT
```

Output

```
[528] List '0' cleared.
```

Save the list:

jsh

```
SELECT F.COMPANY
SAVE.LIST BRANCHES
```

Output

```
 7 Records selected
 7 record(s) saved to list 'BRANCHES'
```

Retrieve the list:

jsh

```
GET.LIST BRANCHES
```

Output

```
7 Records selected

>
```

Use active SELECT list to display live COMPANY file (e.g.  with descending sort by field MNEMONIC):

jsh

```
LIST F.COMPANY COMPANY.NAME MNEMONIC BY.DSND MNEMONIC
```

Output

```
@ID........    COMPANY.NAME..........    MNEMONIC

  SG0010001    Model Bank - Singapore    SG1
  GB0010004    MF Branch 2               MF3
  GB0010003    MF Branch 1               MF2
  GB0010002    MF LEAD COMPANY           MF1
  EU0010001    Model Bank - Europe       EU1
  GB0010005    Model Bank Branch 1       BR1
  GB0010001    Model Bank R14            BNK
```

Save a field rather than @ID; use T24 users list and see to which branches users are assigned:

jsh

```
SELECT F.USER SAVING COMPANY.CODE
```

Output

```
 203 Records selected
```

Remove duplicate results (use CLEARSELECT before it to get rid of the previous example result):

jsh

```
SELECT F.USER SAVING UNIQUE COMPANY.CODE
```

Output

```
 8 Records selected
```

Use SELECT list to list another file:

jsh

```
SELECT F.USER SAVING UNIQUE COMPANY.CODE
LIST ONLY F.COMPANY
```

Output

```
@ID........

GB0010001
 ** Error [ 202 ] **
Record 'ALL' is not on file.
GB0010002
GB0010003
GB0010004
SG0010001
EU0010001
GB0010005
```

*(Not necessarily the field contains a valid COMPANY code - see “ALL”-related error above.)*

Save an expression using string concatenation:

jsh

```
SELECT F.COMPANY SAVING EVAL "MNEMONIC:': ':COMPANY.NAME"
SAVE.LIST BRANCHES
CT &SAVEDLISTS& BRANCHES
```

Output

```
    BRANCHES
001 MF2: MF Branch 1
002 BR1: Model Bank Branch 1
003 SG1: Model Bank - Singapore
004 EU1: Model Bank - Europe
005 MF1: MF LEAD COMPANY
006 MF3: MF Branch 2
007 BNK: Model Bank R14
```

## 1.21 Read a record in a program (jBASE style)

Edit PROG.0 contents to output the T24 release:

jBC

```
$INSERT I_F.SPF

OPEN 'F.SPF' TO f_spf ELSE ABORT 201, 'F.SPF'
READ rec_spf FROM f_spf, 'SYSTEM' ELSE
   CRT 'Read error'
   STOP
END

t24_rel = rec_spf<Spf_CurrentRelease>

CRT 'Right Ho, T24!', t24_rel, 'detected'
```

Angle brackets that are used in “rec_spf<Spf_CurrentRelease>” refer to the particular element of a dynamic array. READ returns the full record as a dynamic array and Spf_CurrentRelease is defined in I_F.SPF:

jsh

```
CT T24.BP I_F.SPF
```

Output

```
001 * File Layout for SPF Created 17 MAR 10 at 12:44PM by sasireka
002 *      PREFIX[SPF.]      SUFFIX[]
003          EQU SPF.RUN.DATE TO 1,  Spf_RunDate TO 1,
004             SPF.SITE.NAME TO 2, Spf_SiteName TO 2,
005               SPF.OP.MODE TO 3, Spf_OpMode TO 3,
006            SPF.OP.CONSOLE TO 4, Spf_OpConsole TO 4,
007          SPF.MAIN.ACCOUNT TO 5, Spf_MainAccount TO 5,
008        SPF.BACKUP.CYCLE.1 TO 6, Spf_BackupCycle1 TO 6,
009        SPF.BACKUP.CYCLE.2 TO 7, Spf_BackupCycle2 TO 7,
010       SPF.CURRENT.RELEASE TO 8, Spf_CurrentRelease TO 8,        <=====
...
```

So this field has #8 and we can list it using its name (without prefix)...

jsh

```
LIST F.SPF CURRENT.RELEASE
```

...or its number (using so-called “A-correlative”, same as we used before for listing VOC):

jsh

```
LIST F.SPF *A8
```

Output:

```
@ID...    *A8..........

SYSTEM    R14
```

Compile the program (we need T24 core application insert file I_F.SPF hence “-I”):

jsh

```
BASIC -I../T24_BP ETC.BP PROG.0
CATALOG ETC.BP PROG.0
```

Run:

jsh

```
PROG.0
```

Output

```
Right Ho, T24!  R14     detected
```

To output the line more neatly we can use the concatenation of strings:

jBC - source of PROG.0

```
CRT 'Right Ho, T24! ' : t24_rel : ' detected'
```

Output

```
Right Ho, T24! R14 detected
```

## 1.22 Retrieve multi-values

Again amend the program PROG.0 - add some code to the end:

jBC - old code

```
$INSERT I_F.SPF

OPEN 'F.SPF' TO f_spf ELSE ABORT 201, 'F.SPF'
READ rec_spf FROM f_spf, 'SYSTEM' ELSE
   CRT 'Read error'
   STOP
END

t24_rel = rec_spf<Spf_CurrentRelease>

CRT 'Right Ho, T24!', t24_rel, 'detected'
```

jBC - changes

```
* new contents below: see products list
prod_list = rec_spf<Spf_Products>

* result is also a dynamic array with one field and many values
CRT 'Products found: ' : DCOUNT(prod_list, @VM)                ;* @VM is value mark
CRT 'Products are: ' : CHANGE(prod_list, @VM, ', ')
```

Additional output

```
Products found: 95
Products are: AA, AB, AC, AD, AI, AL, AM, AP, AR, BE, BL, BM, BR, CM, CO, CR, DC, DD,
DE, DL, DM, DS, DW, DX, EB, ET, EU, FA, FD, FE, FR, FT, FW, FX, GP, IA, IB, IC, IF,
IM, IN, IX, LC, LD, LI, LM, MC, MD, MF, MM, MO, MS, MT, NR, NS, OF, OO, OP, OV, PC,
...
```

Before screen output each of non-printable delimiters (@VM is ASCII 253) was replaced to comma and a space.  So dynamic array can be considered as a string with 3 types of special delimiters (the other 2 are:  @FM - field mark - ASCII 254 and @SM - subvalue mark - ASCII 252).

To format the long line of products we can use FOLD() function.  In the following example @FM delimiters that FOLD() uses in its result will be replaced to end-of-line characters:

jBC

```
...
prod_list = rec_spf<Spf_Products>
CRT 'Products found: ' : DCOUNT(prod_list, @VM)
CHANGE @VM TO ', ' IN prod_list
CRT 'Products are:'
CRT CHANGE(FOLD(prod_list, 50), @FM, CHAR(10))
```

Formatted output

```
Products are:
AA, AB, AC, AD, AI, AL, AM, AP, AR, BE, BL, BM,
BR, CM, CO, CR, DC, DD, DE, DL, DM, DS, DW, DX,
EB, ET, EU, FA, FD, FE, FR, FT, FW, FX, GP, IA,
IB, IC, IF, IM, IN, IX, LC, LD, LI, LM, MC, MD,
MF, MM, MO, MS, MT, NR, NS, OF, OO, OP, OV, PC,
PD, PM, PO, PP, PR, PV, PW, RC, RE, RP, RS, SA,
SB, SC, SE, SF, SG, SL, SP, ST, SW, SY, T2, T3,
T4, T5, TK, TT, TX, VL, VP, VS, WR, WS, XT
```

Note about errors handling.  Though the THEN or ELSE statement is mandatory for OPEN, it’s tempting to use “ELSE NULL” there - there’s always F.SPF, isn’t it?  Well, there can be many reasons why this file wouldn’t open - network error, RDBMS driver problem, even a typo in the program.  Error conditions should always be checked.

In the following code there’s an error in file name:

jBC

```
OPEN 'F.Spf' TO f_spf SETTING ret_code ELSE NULL
CRT ret_code
READ rec_spf FROM f_spf, 'SYSTEM' ELSE
   CRT 'Read error'
   STOP
END
```

Output

```
128
 ** Error [ NOT_FILE_VAR ] **
Variable is not an opened file descriptor ,
Var f_spf , Line     3 , Source PROG.1
Trap from an error message, error message name = NOT_FILE
Source changed to D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\ETC.BP\PROG.1
0003 READ rec_spf FROM f_spf, 'SYSTEM' ELSE
jBASE debugger->
```

jBASE is case-sensitive so F.Spf couldn’t be found, hence the error code 128.  To leave the debugger use the command Q.

## 1.23  First T24 subroutine

In T24 all non-core logic (called “local development”) is done via subroutines that are launched by T24 core at certain events like record @ID input, commit, authorisation etc.  The following example shows “mainline” routine that can be launched at T24 “AWAITING APPLICATION” prompt (only in terminal mode though).  This subroutine will output to the screen the T24 maintenance date (the date after which a new maintenance code is to be obtained from Temenos).

Create a subroutine SUBR.0 in ETC.BP (checking that the chosen name isn’t used):

jsh

```
jshow -c SUBR.0
```

No output, so:

jsh

```
JED ETC.BP SUBR.0
```

The source code will be:

jBC

```
SUBROUTINE SUBR.0
*-----------------------------------------------------------------------
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_F.SPF
*-----------------------------------------------------------------------
* We don't need to read F.SPF as before - it's already done by the core;
* record contents were placed into global variable R.SPF.SYSTEM
* declared in I_COMMON:

   maint_date = R.SPF.SYSTEM<Spf_MaintenanceDate>
// don't use CRT or PRINT here - it will corrupt the screen:
   TEXT = 'Maintenance date: ' : maint_date
   CALL REM

   RETURN
END
```

After copy/paste the screen looks like:

JED

```
*File ETC.BP , Record 'SUBR.0'                              Insert      11:42:41
Command->
0001 SUBROUTINE SUBR.0
0002 *-----------------------------------------------------------------------
0003 $INSERT I_COMMON
0004 $INSERT I_EQUATE
0005 $INSERT I_F.SPF
0006 *-----------------------------------------------------------------------
0007 * We don't need to read F.SPF as before - it's already done by the core;
0008 * record contents were placed into global variable R.SPF.SYSTEM
0009 * declared in I_COMMON:
0010 maint_date = R.SPF.SYSTEM<Spf_MaintenanceDate>
0011 // don't use CRT or PRINT here - it will corrupt the screen:
0012 TEXT = 'Maintenance date: ' : maint_date
0013 CALL REM
0014 RETURN
0015 END
------------------------------- End Of Record -------------------------------
```

Press Esc to go to command line, then enter the command BI.

The source code is now formatted:

JED

```
*File ETC.BP , Record 'SUBR.0'                              Insert      11:43:26
Command->
0001     SUBROUTINE SUBR.0
0002 *-----------------------------------------------------------------------
0003     $INSERT I_COMMON
0004     $INSERT I_EQUATE
0005     $INSERT I_F.SPF
0006 *-----------------------------------------------------------------------
0007 * We don't need to read F.SPF as before - it's already done by the core;
0008 * record contents were placed into global variable R.SPF.SYSTEM
0009 * declared in I_COMMON:
0010     maint_date = R.SPF.SYSTEM<Spf_MaintenanceDate>
0011     // don't use CRT or PRINT here - it will corrupt the screen:
0012     TEXT = 'Maintenance date: ' : maint_date
0013     CALL REM
0014     RETURN
0015 END
------------------------------- End Of Record -------------------------------
```

Add an empty line at the end, save and compile.

List ETC.BP after compilation:

jsh

```
LIST ETC.BP
```

Output

```
...
$SUBR.0
SUBR.0
...
```

$SUBR.0 is the object module that isn’t necessary after compilation, as it was for PROG.0 earlier.

See where the executables are:

jsh

```
LIST lib
```

Output

```
.jbase_lock_file
jLibDefinition
lib0.dll
lib1.dll
lib2.dll
lib3.dll
lib3_TMP_0.dll
libdef.def
objdir
```

jLibDefinition sets compilation options:

jsh

```
CT lib jLibDefinition
```

Output

```
...
043 ##
044 libname = lib%n.dll
045 exportname = libdef.def
046 objectdir = objdir
047 maxsize = 8M
048 baseaddress = 768M
```

We don’t need to amend these settings.  If this file is accidentally deleted, it will be recreated during the next compilation of any subroutine.

“objdir” is the directory where object modules are copied.  Don’t delete them since they are used whenever a library is rebuilt:

jsh

```
LIST lib\objdir
```

Output

```
...
SUBR_2E0.obj         <=== exact copy of $SUBR.0
...
```

*(Dot in subroutine name was replaced to “_2E”; other characters like underscores are replaced as well.)*

Creation of executables for subroutines in “lib” is stipulated by JBCDEV_LIB environment variable.  This directory also has to be mentioned in the environment variable JBCOBJECTLIST:

jsh

```
echo %JBCDEV_LIB%
```

Output

```
D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\lib
```

jsh

```
jdiag -v
```

Output

```
...
Subroutine dir (Default)    : 'D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\lib'
Subroutine path 'D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\lib'
is in JBCOBJECTLIST
...
```

See where exactly SUBR.0 is:

jsh

```
jshow -c SUBR.0
```

Output

```
Subroutine:     D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\lib\lib3.dll
                jBC SUBR.0 version 14.0 Mon Dec 15 16:28:11 2014
                jBC SUBR.0 source file ETC.BP
```

See lib3.dll full contents:

jsh

```
jshow -a lib3.dll
```

Output

```
...
JBC_GPAC_2EMDC_2EPARAMTER_2EVALIDATE
JBC_SUBR_2E0
JBC_V_2EASSET_2EREGISTER_2EAUTH
...
```

If our subroutine is copied to another directory, amended and compiled there - our version of subroutine won’t be visible to jBASE and therefore to T24.  If there are different “BP” directories in development environment, it might happen that certain subroutine or program presents in more that one of them.  Two people having two different versions of a subroutine will overwrite each other’s work.  So before creating a subroutine or finding the latest copy of an existing one, always use jshow command.

If “lib” directory is located before “t24lib” in JBCOBJECTLIST (as in our case), local routine with the same name as a core one will always be executed and a core one will never be; that is absolutely wrong.

For example, we accidentally created a local subroutine CURRENCY, like:

jBC

```
   SUBROUTINE CURRENCY(ccy_code)
$INSERT I_COMMON
$INSERT I_EQUATE

   TEXT = 'Currency is ' : ccy_code
   CALL REM
   RETURN
END
```

Then, after its compilation, launch T24 and enter CURRENCY at “AWAITING APPLICATION” prompt:

T24

```
 -- LAST SIGN.ON, DATE: 06 FEB 2015     TIME: 13:33        ATTEMPTS: 0 --------
 06 MAR 2015 09:46:29  USER (22 APR) VLADIMIR.K        [22048,IN]
 ACTION  ** Error [ SUBROUTINE_PARM_ERROR ] **
'SUBROUTINE CURRENCY' called with incorrect arguments , Line     1 ,
                                                    Source EB.EXECUTE.APPLICATION
Trap from an error message, error message name = SUBROUTINE_PARM_ERROR
Line 1 , Source EB.EXECUTE.APPLICATION
jBASE debugger->
```

See jshow output:

jsh

```
jshow -c CURRENCY
```

Output

```
Subroutine:          D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\lib\lib3.dll
                     jBC CURRENCY version 14.0 Fri Mar 06 13:45:52 2015
                     jBC CURRENCY source file ETC.BP
Subroutine (DUP!!):  D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\t24lib\
                            fin_currencyconfig\lib0.dll
                     jBC CURRENCY version 14.0 Wed May 14 16:56:17 2014
                     jBC CURRENCY source file source/r14/WIN64_TAFCR14GA
```

To correct the situation, use DECATALOG utility:

jsh

```
DECATALOG ETC.BP CURRENCY
```

Output

```
Object CURRENCY decataloged successfully
Library D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\lib\lib3.dll rebuild okay
```

See jshow output again:

jshow output

```
Subroutine:          D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\t24lib\
                            fin_currencyconfig\lib0.dll
                     jBC CURRENCY version 14.0 Wed May 14 16:56:17 2014
                     jBC CURRENCY source file source/r14/WIN64_TAFCR14GA
```

Now T24 shows the correct screen - one of CURRENCY application:

T24

```
 Model Bank R14          CURRENCY


 ------------------------------------------------------------------------------









 -- LAST SIGN.ON, DATE: 06 MAR 2015     TIME: 09:46        ATTEMPTS: 0 --------
 06 MAR 2015 10:53:51  USER (22 APR) VLADIMIR.K        [16045,IN]
 ACTION _
 AWAITING FUNCTION
```

Back to SUBR.0.  To use this subroutine as a template, copy it using another name, e.g.:

jsh

```
COPY FROM ETC.BP SUBR.0,SUBR.T
```

*(No space after comma!)*

Output

```
1 records copied
```

*(Don’t forget to change subroutine name in the source code!)*

To be able to run SUBR.0 from T24 “AWAITING APPLICATION” prompt we need to create PGM.FILE record.  Log in to T24, type PGM.FILE, I SUBR.0.  Populate 2 fields:

T24

```
Model Bank R14          PROGRAM FILE, INPUT

    PROGRAM            SUBR.0
--------------------------------------------------------------------------------
  1 TYPE.............. M
  2. 1 GB SCREEN.TITLE
  3 ADDITIONAL.INFO...
  4. 1 BATCH.JOB......
  5 PRODUCT........... EB                  System Core
  6 SUB.PRODUCT.......
  7. 1 DESCRIPTION....
  8. 1 APPL.FOR.SUBR..
  9 ACTIVATION.FILE...
 10 MT.KEY.COMPONENT..
 11 MT.KEY.FILE.......
 12 REC.VERIFY........
 13 BYPASS.SEL........
 14 BULK.NO...........
 15 JOB.RATING........
 16 RESERVED.9........
--------------------------------------------------------------------------------
07 OCT 2014 13:26:56  USER (22 APR) INPUTTER          [22945,IPAGE 1   >>>3>>>
ACTION
```

Commit.

*“System Core” to the right of “EB” is so-called “enrichment” that the core outputs if there’s another table or values list assigned to a field.*

Type SUBR.0:

T24

```
--------------------------------------------------------------------------------
07 OCT 2014 13:32:44  USER (22 APR) INPUTTER          [22945,IN]
ACTION
CONTINUE (Y)                                      Maintenance date: 20150630
```

Date here is being output as it is stored in data file.

To enhance the output use ICONV/OCONV functions:

jBC - changes

```
   maint_date = R.SPF.SYSTEM<Spf_MaintenanceDate>
   internal_date = ICONV(maint_date, 'D')

   * calculate the difference
   the_diff = internal_date - DATE()

   out_date = OCONV(internal_date, 'D')
   TEXT = 'Maintenance date: ' : out_date : ' - '   \
        : the_diff :  ' days to go'
   CALL REM
```

*(Backslash signifies line continuation.)*

Output

```
-- LAST SIGN.ON, DATE: 07 OCT 2014     TIME: 13:26        ATTEMPTS: 1 --------
07 OCT 2014 14:30:40  USER (22 APR) INPUTTER          [12025,IN]
ACTION
CONTINUE (Y)                    Maintenance date: 30 JUN 2015 - 266 days to go
```

## 1.24 Other date and time manipulations

jBC program

```
CRT OCONV(1, 'D')             ;* 01 JAN 1968
CRT OCONV(DATE(), 'D')        ;* here and below output for 07 OCT 2014
CRT OCONV(DATE(), 'D2')       ;* 07 OCT 14
CRT OCONV(DATE(), 'D4/')      ;* 10/07/2014
CRT OCONV(DATE(), 'DY')       ;* 2014
CRT OCONV(DATE(), 'DY2')      ;* 14
CRT OCONV(DATE(), 'DQ')       ;* 4 (quarter)
CRT OCONV(DATE(), 'DM')       ;* 10 (month number)
CRT OCONV(DATE(), 'DMA')      ;* OCTOBER
CRT OCONV(DATE(), 'DD')       ;* 7
CRT OCONV(DATE(), 'DJ')       ;* 280 (number of a day in the year)
CRT OCONV(DATE(), 'DW')       ;* 2 (day number in a week, starting from Monday)
CRT OCONV(DATE(), 'DWA')      ;* TUESDAY
CRT OCONV(TIME(), 'MT')       ;* 15:21
CRT OCONV(TIME(), 'MTS')      ;* 15:21:17
CRT OCONV(1, 'MTS')           ;* 00:00:01
CRT SYSTEM(12)                ;* 55277906.2495 - number of milliseconds past midnight
```

Double-check the output on the last line in jQL:

jsh

```
LIST . SAMPLE 1 EVAL "SYSTEM(12)/1000/3600" AS Hours ID.SUPP
```

Output

```
Hours.....

15.3533
```

Or - use division remainder to get the proper time output:

jsh

```
LIST . SAMPLE 1 EVAL "INT(SYSTEM(12)/1000/3600):':'
   :INT(MOD(SYSTEM(12)/1000/60,60))" AS TimeNow ID.SUPP
```

Output

```
TimeNow...

15:21
```

*(All jQL queries should be on the same line; there’s no line continuation symbol available.)*

## 1.25 Read a record (T24 style)

Amend the subroutine SUBR.0 the following way:

jBC

```
SUBROUTINE SUBR.0
*-----------------------------------------------------------------------
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_F.CUSTOMER
*-----------------------------------------------------------------------

* Use OPF() to open a file - it maintains the pool for opened files.
* We  don't need to know the file prefix - OPF will find it out.

   fn_cust = 'F.CUSTOMER'  ;  f_cust = ''
   CALL OPF(fn_cust, f_cust)

* Now we know full name - it's in fn_cust.
* To get some @ID we can use Basic SELECT.
* It does not execute the full SELECT as jQL but sets sort of a pointer that
* can be used to move along the file.
* New @IDs that would be added after basic SELECT might be retrieved as well -
* but for jQL SELECT the full list is obtained immediately as a snapshot
* (and therefore it's slower).

   SELECT f_cust TO id_list

* Now get the first one

   READNEXT cust_id FROM id_list THEN
```

```
      CALL F.READ(fn_cust, cust_id, rec_cust, f_cust, '')
      short_name = rec_cust<Customer_ShortName>  ;* can also use <EB.CUS.SHORT.NAME>
      READNEXT next_id FROM id_list THEN NULL ELSE next_id = 'not found'

* Now use special message format with placeholders.
* It's crucial for non-English language environments.

      TEXT = 'Customer &: short name "&", next: &' :FM: cust_id   \
             :VM: short_name :VM: next_id
      CALL REM
   END

   RETURN
END
```

*(FM is the same as @FM - it's assigned via EQU[ATE] in I_EQUATE; same goes for VM and @VM, SM and @SM respectively.)*

Output

```
 -- LAST SIGN.ON, DATE: 07 OCT 2014     TIME: 16:36        ATTEMPTS: 0 --------
 07 OCT 2014 16:38:43  USER (22 APR) INPUTTER          [8911,IN]
 ACTION
 CONTINUE (Y) Customer 188888: short name "BBH Inhouse Custodian", next: 122122
```

*(After SELECT @IDs are not sorted; use SSELECT if you need that.)*

## 1.26 T24 "concat" files

Files that are not applications but are linked to them and act like indexes are called "concat" files. For example, how to obtain the list of all accounts for particular customer:

jsh

```
LIST ONLY FBNK.ACCOUNT WITH CUSTOMER EQ 188888
```

Output

```
@ID.............

66435
66408

 2 Records Listed
```

This could be very long - depending on records number... which is:

jsh

```
COUNT FBNK.ACCOUNT
```

Output

```
 1385 Records counted
```

Not much but it’s a demo database.  Proper method is to access a concat file by @ID:

jsh

```
CT FBNK.CUSTOMER.ACCOUNT '188888'
```

Output

```
    188888
001 66408
002 66435
```

*(List of concat files isn’t documented; when user screen for particular application - VERSION - is activated, list of corresponding concats can be found in T24 global array CONCATFILE.)*

T24 core updates concat files automatically.  Let’s take USER record TRAIN511 with sign on name TRAIN51:

jsh

```
LIST F.USER 'TRAIN511'
```

Output

```
@ID................. TRAIN511
USER.ID............. TRAIN511
USER.NAME........... TRAIN5111
SIGN.ON.NAME........ TRAIN51
CLASSIFICATION...... INT
...
```

jsh

```
LIST F.USER.SIGN.ON.NAME 'TRAIN51'
```

Output

```
@ID.............   @ID.............   USER.SIGN.ON.NAME   USER.ID.........

TRAIN51            TRAIN51            TRAIN51             TRAIN511

 1 Records Listed
```

Log in to T24, amend this record - set SIGN.ON.NAME to TRAIN5123.  F7 key (or Ctrl-T Enter) in a field puts us into edit mode (indicated by the pipe at the end of the field):

T24

```
Model Bank R14          USER PROFILE, INPUT

    USER.ID........... TRAIN511
------------------------------------------------------------------------------
  1 USER.NAME......... TRAIN5111
  2 SIGN.ON.NAME...... TRAIN51         |
  3 CLASSIFICATION.... INT
  4 LANGUAGE.......... 1                    English
  5. 1 COMPANY.CODE... GB0010001            Model Bank R14
  5. 2 COMPANY.CODE... GB0010002            MF LEAD COMPANY
  5. 3 COMPANY.CODE... EU0010001            Model Bank - Europe
  5. 4 COMPANY.CODE... SG0010001            Model Bank - Singapore
  6 DEPARTMENT.CODE... 1                    Implementation
  7 PASSWORD.VALIDITY. 01 SEP 2014 M0601    01 SEP 2014 Every 6 months on day 1
  8 START.DATE.PROFILE 28 MAR 2014
  9 END.DATE.PROFILE.. 28 MAR 2020
 10. 1 START.TIME..... 00:00
 11. 1 END.TIME....... 24:00
 12 TIME.OUT.MINUTES.. 999
 13 ATTEMPTS.......... 9
------------------------------------------------------------------------------
15 DEC 2014 09:51:12  USER (22 APR) INPUTTER          [792,IN]PAGE 1   >>>6>>>
ACTION
```

Press F3 several times to move forward, type 23 at the end.  To exit the edit mode press F7 again:

T24

```
Model Bank R14          USER PROFILE, INPUT

    USER.ID........... TRAIN511
------------------------------------------------------------------------------
  1 USER.NAME......... TRAIN5111
  2 SIGN.ON.NAME...... TRAIN5123
  3 CLASSIFICATION.... INT
  ...
```

*(If we put there an existing sign on name, e.g.  INPUTT, we'll get the field validation error "ALREADY STORED IN F.USER.SIGN.ON.NAME".)*

Commit the record and see concat being updated:

jsh

```
LIST F.USER.SIGN.ON.NAME 'TRAIN5123'
```

Output

```
@ID.............   @ID.............   USER.SIGN.ON.NAME   USER.ID.........

TRAIN5123          TRAIN5123          TRAIN5123           TRAIN511

 1 Records Listed
```

Old record doesn’t exist anymore:

jsh

```
LIST F.USER.SIGN.ON.NAME 'TRAIN51'
```

Output

```
 No Records Listed
```

*(In edit mode you can also use:  F1 - cancel, F2 - one character back, F4 - end, F5 - delete a character, F6 - toggle insert/overwrite; overwrite is default mode.)*

## 1.27 Patterns matching

In jQL there are keywords LIKE and UNLIKE (we already used LIKE for VOC listing); in jBC there’s statement MATCHES. Examples:

List @IDs that end with “01”:

jsh

```
LIST ONLY F.COMPANY WITH @ID LIKE "...'01'"
```

Output

```
@ID........

SG0010001
EU0010001
GB0010001
```

*(We use single quotes inside the expression to define a string constant.  Some patterns look like strings so we are on the safe side here.)*

List @IDs that start with “GB” followed by 7 numeric characters:

jsh

```
LIST ONLY F.COMPANY WITH @ID LIKE "'GB'7N"
```

Output

```
@ID........

GB0010003
GB0010005
GB0010002
GB0010004
GB0010001
```

Why single quotes matter:

jsh

```
LIST FBNK.CUSTOMER STREET WITH STREET LIKE "'46A'..."
```

Output

```
@ID.......    STREET.............................

    129028    46A Bridge Avenue
```

jsh

```
LIST FBNK.CUSTOMER STREET WITH STREET LIKE "46A..."
```

Output

```
@ID.......    STREET.............................

 No Records Listed
```

*(Pattern 46A without quotes means 46 alpha characters.)*

List streets that have the space at position 6:

jsh

```
LIST FBNK.CUSTOMER STREET WITH STREET LIKE "5X' '..."
```

Output

```
@ID.......    STREET.............................

    100999    DELTA STREET
    100473    TOWER BUILDINGS 1002
    111613    24133 NORTHWESTERN HIGHWAY
              SOUTHFIELD
    111660    435 ORCHARD ROAD 04-01
              WISMA ATRIA SINGAPORE 238877
    100269    6 735 KITASHINAGAWA
    100310    39 W. LEXINGTON STREET
    100600    857/S II
    100254    CABLE HOUSE 54 - 62 NEW BROAD ST
    129012    6 Rue Godefroy puteaux 92800
    ...
```

*(“X” means “any” character; 0 in quantity means “any number of characters” so “0X” is the same as “...”.)*

Why 111660 is there?  Because STREET is a multi-valued field and the second value (“WISMA ATRIA SINGAPORE 238877”) satisfies our selection criterion.  See the record in JED (yes, we can use it to edit data records also):

jsh

```
JED FBNK.CUSTOMER 111660
```

JED

```
File FBNK.CUSTOMER , Record '111660'                        Insert      17:08:01
Command->
0001 SONYSG
0002 SONY SINGAPORE
0003 Sony Singapore Pte Ltd
0004 Sony Singapore Pte Ltd
0005 435 ORCHARD ROAD 04-01]WISMA ATRIA SINGAPORE 238877
0006
0007 SINGAPORE
0008
```

“]” in the line 0005 (which is actually field 5) only looks like a regular character but it’s really a @VM.

To exit without saving, Press Esc (which serves as a toggle between record body and command line) and type EX command.

*(To input @VM instead of right square bracket in editor, use Ctrl-]; for @SM it’s Ctrl-\.)*

Pressing Enter key accidentally in the record body adds one more field and will ruin the record structure.  Let’s add an empty line for that CUSTOMER record between fields 3 and 4, save it and then open it in T24, function I:

T24 - corrupted record

```
Model Bank R14          CUSTOMER INPUT

    CUSTOMER.CODE..... 111660
--------------------------------------------------------------------------------
  1 MNEMONIC.......... SONYSG
  2. 1 GB SHORT.NAME.. SONY SINGAPORE
  3. 1 GB NAME.1...... Sony Singapore Pte Ltd
  4. 1 GB NAME.2......
  5. 1 GB STREET...... Sony Singapore Pte Ltd
  6. 1. 1 GB ADDRESS.. 435 ORCHARD ROAD 04-01
  6. 2. 1 GB ADDRESS.. WISMA ATRIA SINGAPORE 238877
  7. 1 GB TOWN.COUNTRY
  8. 1 GB POST.CODE... SINGAPORE
  9. 1 GB COUNTRY.....
 10. 1 RELATION.CODE..
 11. 1 REL.CUSTOMER...
 12. 1 REVERS.REL.CODE
 13. 1. 1 REL.DELIV.OP
 14. 1. 1 ROLE........
 15. 1. 1 ROLE.MORE.IN
--------------------------------------------------------------------------------
03 FEB 2015 11:45:01  USER (22 APR) TRAIN.01          [3066,INPAGE 1   >>12>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

The value of GB TOWN.COUNTRY field moved down.  Go to page 12 - the last page:

T24 - corrupted record

```
Model Bank R14          CUSTOMER INPUT

    CUSTOMER.CODE..... 111660
--------------------------------------------------------------------------------
176 RESERVED.03.......
177 RESERVED.02.......
178 RESERVED.01.......
179. 1 SEGMENT........
179. 2 CU.EFF.DATE....
179. 3 WELCOME.PACK...
180. 1 OVERRIDE.......
181 RECORD.STATUS.....
182 CURR.NO...........
183. 1 INPUTTER....... 2
184. 1 DATE.TIME...... S_  ER__ SE:AT
185 AUTHORISER........ 1405151649
186 CO.CODE........... _O-FS_-SEAT
187 DEPT.CODE......... GB0010001           MISSING DEPT.ACCT.OFFICER - RECORD
188 AUDITOR.CODE...... 1
189 AUDIT.DATE.TIME...
--------------------------------------------------------------------------------
```

Don’t commit; go back to editor and correct the record.  Such records can appear after incorrect conversion of tables during T24 release upgrade.

Back to patterns matching.  If we want all values to satisfy the pattern - EVERY keyword helps:

jsh

```
LIST FBNK.CUSTOMER STREET WITH EVERY STREET LIKE "5X' '..."
```

Output

```
@ID.......    STREET.............................

    100999    DELTA STREET
    100473    TOWER BUILDINGS 1002
    100269    6 735 KITASHINAGAWA
    100310    39 W. LEXINGTON STREET
    100600    857/S II
    100254    CABLE HOUSE 54 - 62 NEW BROAD ST
    129012    6 Rue Godefroy puteaux 92800
    129026    Tower Heights
    100361    SUGAR QUAY LOWER THAMES ST
    111566    19001 South Western Ave
    101002    DELTA STREET
    128023    56-64 Leonard St
    111560    Obama Avenue
    100258    20555 SH 249
    128078    617 E Cooper Avenue
    100353    5-2-1 Ginza
    ...
```

See that record 111660 is no more here.  To be absolutely sure, add its @ID to selection criteria:

jsh

```
LIST FBNK.CUSTOMER STREET WITH EVERY STREET LIKE "5X' '..." AND @ID EQ 111660
```

Result is “No Records Listed”.

## 1.28  Introduction to locks

As we opened the CUSTOMER record in JED in the previous chapter, jBASE applied a lock to this record.  Let’s open it again and in another session see it:

Session 1

```
JED FBNK.CUSTOMER 111660
```

Session 2 - jsh

```
SHOW-ITEM-LOCKS
```

(*This command could take some time if there are some defunct ports.)*

Output

```
   PORT        PID FILENAME          RECORDKEY         LOCK#         PORT/-PID

      2       2032 ..\bnk.data\st\FB 111660            0x055c33da,W        ---
                   NK_CUSTOMER
```

If we try to JED the same record in session 2 - it will be opened in READONLY mode:

JED

```
Record '111660' is locked by another process
Setting READ ONLY mode - no updates allowed
```

More about locks later.

## 1.29 What is “version” in T24 context?

Despite the name, “version” is a customized user screen. All of them are stored in T24 application VERSION. @ID consists of application @ID, comma and optional user-defined name, e.g. CUSTOMER,AMEND.

“Comma VERSIONs” (without name after comma) are usually used for “raw” input - without customization - and with zero authorisation. We used “PGM.FILE,” VERSION already to create the record for SUBR.0 - so comma wasn’t just part of syntax.

## 1.30 Application record life cycle

Regular VERSIONs are used for user input, after that application record typically obtains status INAU (or CNAU if the record was copied from another one) and is saved to NAU file.

Then another user has to authorise the record and then it’s removed from NAU file and is saved to LIVE one.

Before saving, user can put a record to “HOLD”. In this case the record is saved to NAU file with status of IHLD or CHLD.

When the existing record is amended, 2 records exist at the same time - one in NAU and another in LIVE file.

When amended record is committed, the record is updated in LIVE file and the old record is moved to HIS file with @ID that has an appendix at the end:

jsh

```
LIST ONLY FBNK.ACCOUNT$HIS SAMPLE 5
```

Output

```
   @ID.................

69612;3
71412;1
USD140100001;1
USD141850001;2
USDGBP190210001;1
```

If a record is in NAU or HLD status, it can be deleted (function D).

If a record was authorised, it can be reversed (function R). Reversal also requires authorisation (status RNAU is assigned first).

Reversed record is moved to HIS file and no more presents in LIVE one.

Reversed record can be restored using “H” function.

*Not all applications allow input (PGM.FILE type L ones do not).*

*Not all applications have history (PGM.FILE type U ones do not).*

*Other limitations may apply depending on business logic of application (e.g. HOLD might be forbidden after first commit or reverse is forbidden altogether).*

## 1.31 First T24 VERSION

Let’s use FUNDS.TRANSFER application (abbreviation FT). Log in to T24, type VERSION, I FT,TRAIN. Populate the following fields, commit:

T24

```
  2 RECORDS.PER.PAGE.. 1
  3 FIELDS.PER.LINE... 1
  ...
 46 NO.OF.AUTH........ 1
```

*(Field 46 will also be populated by default with the value 1; allowed values are 0 - as in “comma” VERSIONs, 1 or 2.)*

Now we can type at T24 “AWAITING APPLICATION” prompt: FT,TRAIN I.

At “AWAITING ID” prompt press F3 to obtain new record @ID automatically:

T24 FT input

```
Model Bank R14          FUNDS.TRANSFER,TRAIN INPUT           REF FT14112R21R6


--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE..
  2 DEBIT.ACCT.NO.....
  3 IN.DEBIT.ACCT.NO..
  4 CURRENCY.MKT.DR... 1                  Currency Market
  5 DEBIT.CURRENCY....
  6 DEBIT.AMOUNT......
  7 DEBIT.VALUE.DATE..
  8 IN.DEBIT.VDATE....
  9 DEBIT.THEIR.REF...
 10 CREDIT.THEIR.REF..
 11 CREDIT.ACCT.NO....
 12 CURRENCY.MKT.CR... 1                  Currency Market
 13 CREDIT.CURRENCY...
 14 CREDIT.AMOUNT.....
 15 CREDIT.VALUE.DATE.
 16 TREASURY.RATE.....
--------------------------------------------------------------------------------
07 OCT 2014 17:33:17  USER (22 APR) INPUTTER          [23733,IPAGE 1   >>14>>>
ACTION
```

*(We see all fields here, same as when using “bare” application.  To see only some fields, define them in VERSION record.)*

Note that some fields were auto-populated by the core.  Press F4 to go to command line, then navigate to the end to see the audit trail:

T24 FT input

```
197 RECORD.STATUS.....
198 CURR.NO...........
199. 1 INPUTTER.......
200. 1 DATE.TIME......
201 AUTHORISER........
202 CO.CODE...........
203 DEPT.CODE.........
204 AUDITOR.CODE......
205 AUDIT.DATE.TIME...
```

Audit trail of new record is empty.

Type HLD to put the record on HOLD, open it again in “See” mode using T24 command FT,TRAIN S FT14112R21R6 - thus we’ll only see populated fields:

T24 FT input

```
Model Bank R14          FUNDS.TRANSFER,TRAIN SEE              REF FT14112R21R6


--------------------------------------------------------------------------------
  4 CURRENCY.MKT.DR... 1                   Currency Market
 12 CURRENCY.MKT.CR... 1                   Currency Market
197 RECORD.STATUS..... IHLD                INPUT Held
199. 1 INPUTTER....... 23733_INPUTTER
200. 1 DATE.TIME...... 07 OCT 14 17:39
202 CO.CODE........... GB-001-0001         Model Bank R14
203 DEPT.CODE......... 1                   Implementation









--------------------------------------------------------------------------------
07 OCT 2014 17:44:17  USER (22 APR) INPUTTER          [32661,IPAGE 1
ACTION
AWAITING PAGE INSTRUCTIONS
```

The field DATE.TIME is represented in readable form but is stored in raw format “YYMMDDHHMM”. See it in NAU file where it’s been saved:

jsh

```
LIST FBNK.FUNDS.TRANSFER$NAU 'FT14112R21R6' DATE.TIME
```

Output

```
@ID.....................    DATE.TIME......

FT14112R21R6                      1410071739
```

So it’s easy to select, say, all unauthorised FTs that were input today...

jsh

```
LIST ONLY FBNK.FUNDS.TRANSFER$NAU WITH DATE.TIME LIKE "'141007'..."
```

...or see how namy of them are already authorised:

jsh

```
SELECT FBNK.FUNDS.TRANSFER$NAU WITH DATE.TIME LIKE "'141007'..."
```

Output

```
 1 Records selected
>
```

Second SELECT is applied with the active SELECT list (hence the prompt change):

jsh

```
>SELECT FBNK.FUNDS.TRANSFER
```

Output

```
 ** Error [ 202 ] **
Record 'FT14112R21R6' is not on file.

 No Records selected
```

No records were authorised yet.

To suppress the error message 202 we can use “(R” option in the second SELECT:

jsh

```
>SELECT FBNK.FUNDS.TRANSFER (R
```

## 1.32 VERSION, auto-assign a field

We can use so-called “AUT.NEW.CONTENT” routine to automatically set some field contents when the record is opened. Firstly, create a subroutine:

jsh

```
jshow -c SUBR.ANC
```

No output, so:

jsh

```
JED ETC.BP SUBR.ANC
```

jBC

```
SUBROUTINE SUBR.ANC

$INSERT I_COMMON
$INSERT I_EQUATE

   IF R.NEW(AF) EQ '' THEN R.NEW(AF) = 'TRAIN DEAL'

   RETURN
END
```

Here we have a global T24 variable AF meaning the current field number (i.e. the field where this routine will be attached to). R.NEW is a global dimensioned array containing the data record that we input. It's allowed to amend the contents of this array in local developments (with caution of course).

Create a technical record in PGM.FILE, type S (@ID = subroutine name):

T24

```
Model Bank R14          PROGRAM FILE, INPUT

    PROGRAM            SUBR.ANC
--------------------------------------------------------------------------------
  1 TYPE.............. S
  2. 1 GB SCREEN.TITLE
  3 ADDITIONAL.INFO...
  4. 1 BATCH.JOB......
  5 PRODUCT........... EB                  System Core
  6 SUB.PRODUCT.......
  7. 1 DESCRIPTION....
  8. 1 APPL.FOR.SUBR.. FUNDS.TRANSFER      FUNDS.TRANSFER
  9 ACTIVATION.FILE...
 10 MT.KEY.COMPONENT..
 11 MT.KEY.FILE.......
 12 REC.VERIFY........
 13 BYPASS.SEL........
 14 BULK.NO...........
 15 JOB.RATING........
 16 RESERVED.9........
--------------------------------------------------------------------------------
07 OCT 2014 18:15:27  USER (22 APR) INPUTTER          [26328,IPAGE 1   >>>3>>>
ACTION
```

Attach to a field in VERSION record:

T24

```
Model Bank R14          VERSION, INPUT

    PGM.NAME.VERSION.. FUNDS.TRANSFER,TRAIN
--------------------------------------------------------------------------------
 49. 1 REKEY.FIELD.NO.
 50. 1 AUTOM.FIELD.NO. DEBIT.THEIR.REF     DEBIT.THEIR.REF
 51. 1 AUT.OLD.CONTENT
 52. 1 AUT.NEW.CONTENT @SUBR.ANC
```

Launch the VERSION again (might need re-login to T24 because of its internal cache). To search for the deal we put to HOLD earlier we can use E E command:

T24

```
-- LAST SIGN.ON, DATE: 07 OCT 2014     TIME: 18:15        ATTEMPTS: 0 --------
07 OCT 2014 18:20:55  USER (22 APR) INPUTTER          [27003,IN]
ACTION FT,TRAIN E E _
AWAITING APPLICATION
```

T24 search screen

```
Model Bank R14          FUNDS.TRANSFER,TRAIN EXCEPTION


--------------------------------------------------------------------------------
    SELECTION FIELDS - EXCEPTION FILE      [EQ NE LT GT LE GE RG NR LK UL]
--------------------------------------------------------------------------------
0.. @ID                   0A. REF.NO                1.. TRANSACTION.TYPE
1A. FT1                   2.. DEBIT.ACCT.NO         3.. IN.DEBIT.ACCT.NO
4.. CURRENCY.MKT.DR       5.. DEBIT.CURRENCY        6.. DEBIT.AMOUNT
7.. DEBIT.VALUE.DATE      8.. IN.DEBIT.VDATE        9.. DEBIT.THEIR.REF
10. CREDIT.THEIR.REF      11. CREDIT.ACCT.NO        12. CURRENCY.MKT.CR
13. CREDIT.CURRENCY       14. CREDIT.AMOUNT         15. CREDIT.VALUE.DATE
16. TREASURY.RATE         17. NEG.DEALER.REFNO      18. PROCESSING.DATE
19. ORDERING.CUST         20. IN.ORDERING.CUS       21. ORDERING.BANK
22. IN.ORDERING.BK        23. ACCT.WITH.BANK        24. ACCT.WITH.BK
25. BEN.ACCT.NO           26. IN.BEN.ACCT.NO        27. BEN.CUSTOMER
28. IN.BEN.CUSTOMER       29. BEN.BANK              30. IN.BEN.BANK
31. CHEQUE.NUMBER         32. PAYMENT.DETAILS       33. IN.PAY.DETAILS
34. BC.BANK.SORT.CODE     35. RECEIVER.BANK         36. REC.CORR.BANK
37. INTERMED.BANK         38. IN.INTMED.BANK        39. MAILING
-- LAST SIGN.ON, DATE: 07 OCT 2014     TIME: 18:15        ATTEMPTS: 0 --------
07 OCT 2014 18:22:00  USER (22 APR) INPUTTER          [27003,INPAGE  1 >>7>>>
ACTION _
ENTER SELECTION (eg CUST EQ 123456) , <F5> TO EXECUTE LIST
```

At ACTION prompt type DATE.TIME LK "’141007’...", press Enter, then F5:

T24

```
Model Bank R14           FUNDS.TRANSFER - Default Exceptions

   ID                         STATUS  DATE.TIME                        FUNCT.
--------------------------------------------------------------------------------
1  FT14112R21R6               IHLD    07 OCT 14 17:39









--------------------------------------------------------------------------------
07 OCT 2014 18:24:42  USER (22 APR) INPUTTER          [27003,IPAGE   1 >>>1>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Keys sequence to edit the record:  1 Enter I Enter F5.  Answer “Y” to “RECORD IN HOLD-STATUS” override query.  See that field DEBIT.THEIR.REF is populated according to the logic of subroutine SUBR.ANC:

T24

```
Model Bank R14          FUNDS.TRANSFER,TRAIN INPUT           REF FT14112R21R6


--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE..
  2 DEBIT.ACCT.NO.....
  3 IN.DEBIT.ACCT.NO..
  4 CURRENCY.MKT.DR... 1                  Currency Market
  5 DEBIT.CURRENCY....
  6 DEBIT.AMOUNT......
  7 DEBIT.VALUE.DATE..
  8 IN.DEBIT.VDATE....
  9 DEBIT.THEIR.REF... TRAIN DEAL
 10 CREDIT.THEIR.REF..
 11 CREDIT.ACCT.NO....
 12 CURRENCY.MKT.CR... 1                  Currency Market
 13 CREDIT.CURRENCY...
 14 CREDIT.AMOUNT.....
 15 CREDIT.VALUE.DATE.
 16 TREASURY.RATE.....
--------------------------------------------------------------------------------
07 OCT 2014 18:26:43  USER (22 APR) INPUTTER          [27003,IPAGE 1   >>14>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

*(If we amend the field DEBIT.THEIR.REF, put record to HOLD again and open it later, out change won’t be overwritten - see subroutine code.)*

This functionality works the same way in Browser:

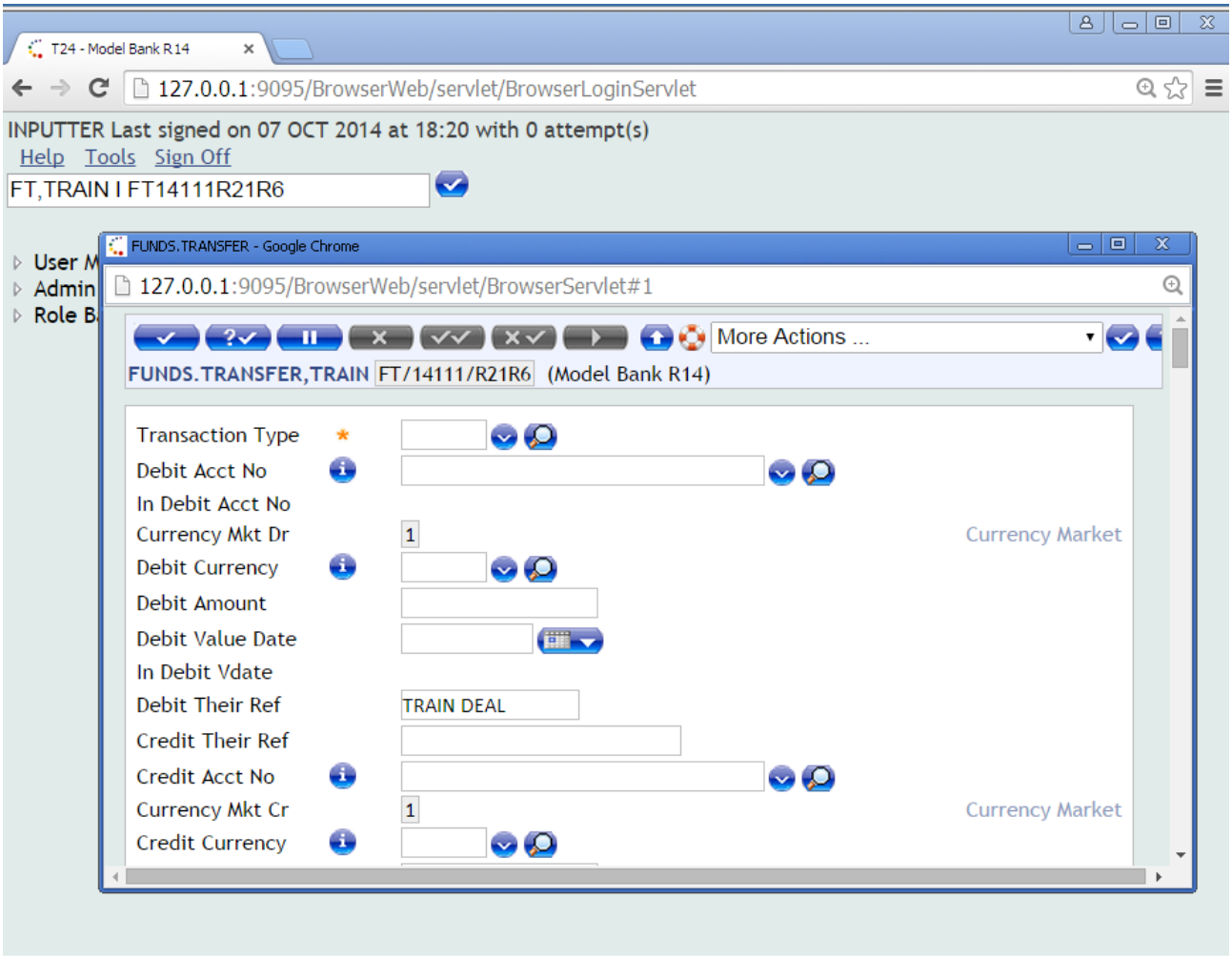


**Figure 18: Same FT version in Browser.**

*(When the record - even not yet existing - is opened in T24 for editing, the lock with its @ID is applied to NAU file.)*

## 1.33 Dimensioned arays

As it was mentioned before, R.NEW is a dimensioned array. One-dimensioned, to be exact so addressing its elements looks like R.NEW(N), where N is a number or a constant that is equated to a number, e.g. R.NEW(9) - which is not recommended - or R.NEW(FundsTransfer_CreditValueDate).

Sample manipulations with dimensioned arrays:

jBC

```
DIM dim_array(10)     ;* declare dimension
MAT dim_array = 0     ;* populate with default value

dim_array(5) = '1'    ;* redefine a value of particular element

FOR i = 1 TO 10
   CRT dim_array(i):  ;* output: 0000100000 (colon at statement end means no new line)
NEXT i
CRT ''                ;* new line finally

* Build a dynamic array from dimensioned one
MATBUILD dyn_array FROM dim_array

dyn_array<5> += 1     ;* "++" doesn't work only here...
dyn_array<7> = 7 :@VM: 8 :@SM: 9

* Elements can be of different types
dyn_array<9> = 'Nine'  ;  dyn_array<10> = 'Ten'

* Output dynamic array in printable form;
* also introducing single- and double-quoting of a string...
CRT SQUOTE(OCONV(dyn_array, 'MCP'))    ;* output: '0^0^0^0^2^0^7]8\9^0^Nine^Ten'

* Build a delimited string from part of dimensioned array
MATBUILD delim_array FROM dim_array, 3, 7 USING ":"
CRT DQUOTE(delim_array)         ;* output: "0:0:1:0:0"

* Back from dynamic to dimensioned array - of lesser size
DIM new_dim_array(8)

MATPARSE new_dim_array FROM dyn_array
CRT new_dim_array(5)        ;* output: 2
CRT CHANGE(new_dim_array(0), @FM, '>>')   ;* output: Nine>>Ten
                                          ;* - elements that didn't fit
```

*(In general, dimensioned arrays are faster to access but dynamic ones are more flexible.)*

## 1.34 VERSION: field validation

To validate a field in VERSION we can use “validation routine”. For example, we need to check if FT debit value date is no more than 5 work days ahead.

Firstly, the source code:

jBC

```
SUBROUTINE SUBR.VAL

$INSERT I_COMMON
$INSERT I_EQUATE

   dr_val_date = COMI                              ;* global var, contains user input
   the_diff = 'W'                          ;* 'W' for work days, 'C' for calendar days
   CALL CDD('', TODAY, dr_val_date, the_diff)        ;* TODAY is another global var;
                                                     ;* never amend it!
   IF the_diff GT 5 THEN ETEXT = "Shouldn't be later than 5 work days ahead"

   RETURN
END
```

*(Double quotes can also be used to define a string; the third way is to use back-slashes.)*

*(There's a lot of global variables in I_COMMON with names like A, E etc; dots are very often used inside a global variable name.  It's a good idea to name your local variables in lowercase since jBC is case-sensitive; an underscore or 2 wouldn't harm as well.)*

Secondly, create a technical record in T24 application EB.API, @ID = routine name:

T24

```
Model Bank R14          EB.API, INPUT

    KEY............... SUBR.VAL
------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION.
  2 PROTECTION.LEVEL.. FULL
  3 SOURCE.TYPE....... BASIC
  4 JAVA.METHOD.......
  5 JAVA.CLASS........
  6 JAVA.PA
  7 RESERVED27........
  8 RESERVED26........
  9 RESERVED25........
 10 RESERVED24........
 11 RESERVED23........
 12 RESERVED22........
 13 RESERVED21........
 14 RESERVED20........
 15 RESERVED19........
 16 RESERVED18........
------------------------------------------------------------------------------
07 OCT 2014 20:57:20  USER (22 APR) INPUTTER          [22662,IPAGE 1   >>>3>>>
ACTION
```

*(Why it was PGM.FILE last time?  EB.API is newer approach and PGM.FILE record is now required only for AUT.NEW.CONTENT routine and couple of other less often used types.)*

Attach the routine to VERSION:

T24

```
Model Bank R14          VERSION, INPUT

    PGM.NAME.VERSION.. FUNDS.TRANSFER,TRAIN
------------------------------------------------------------------------------
 55. 6. 1 VAL.ASSOC... MSG.TYPE
 55. 6. 2 VAL.ASSOC... MSG.DATE
 56. 1. 1 SUB.ASSOC...
 57 LOCAL.REF.FIELD... LOCAL.REF
 58. 1 VALIDATION.FLD. DEBIT.VALUE.DATE    DEBIT.VALUE.DATE         <=====
 59. 1 VALIDATION.RTN. SUBR.VAL                                     <=====
 60. 1 D.SLIP.FORMAT..
 61. 1 D.SLIP.FUNCTION
 62 D.SLIP.TRIGGER....
 63. 1 INPUT.ROUTINE..
 64. 1 AUTH.ROUTINE ..
 65 REPORT.LOCKS...... YES
 66 GTS.CONTROL.......
 67. 1 GB D
 68. 1 ASSOC.VERSION..
 69 NEXT.VE
------------------------------------------------------------------------------
07 OCT 2014 21:02:32  USER (22 APR) INPUTTER          [22662,IPAGE 5   >>>8>>>
ACTION
```

For this routine to work in Browser, DEBIT.VALUE.DATE is to be declared as “Hot field”.  To achieve that it should be explicitly defined in VERSION record:

T24

```
    PGM.NAME.VERSION.. FUNDS.TRANSFER,TRAIN
------------------------------------------------------------------------------
...
 13. 1 FIELD.NO....... DEBIT.VALUE.DATE    DEBIT.VALUE.DATE
...
 42. 1. 1 ATTRIBS.....
```

Field ATTRIBS has a list of choices.  To activate it type “_” - “underscore”, then Enter when the cursor is in this field:

T24

```
------------------------------------------------------------------------------
07 OCT 2014 21:10:36  USER (22 APR) INPUTTER          [22662, PAGE 3   >>10>>>
ACTION
PROPOSAL 'HOT.FIELD' OK. (Y, NO, _)
```

Type Y Enter, then couple of times - F2 so the field will be refreshed (the latter is necessary only for some fields):

T24 VERSION input

```
 42. 1. 1 ATTRIBS..... HOT.FIELD
 ...
```

Other fields will be added as well in this example, see resulting VERSION record:

T24

```
Model Bank R14          VERSION, SEE

    PGM.NAME.VERSION.. FUNDS.TRANSFER,TRAIN
-------------------------------------------------------------------------------
  2 RECORDS.PER.PAGE.. 1
  3 FIELDS.PER.LINE... 1
 13. 1 FIELD.NO....... TRANSACTION.TYPE    TRANSACTION.TYPE
 13. 2 FIELD.NO....... DEBIT.VALUE.DATE    DEBIT.VALUE.DATE
 42. 2. 1 ATTRIBS..... HOT.FIELD
 13. 3 FIELD.NO....... DEBIT.AMOUNT        DEBIT.AMOUNT
```

I function shows that there is a group of fields that are linked together:

T24 VERSION input

```
 13. 2 FIELD.NO....... DEBIT.VALUE.DATE    DEBIT.VALUE.DATE
 14. 2 COLUMN.........
 15. 2 EXPANSION......
 16. 2 TEXT.CHAR.MAX..
 17. 2. 1 TEXT........
 18. 2. 1 TXT.040..078
 19. 2. 1 TXT.079..117
 20. 2. 1 TXT.118..132
 21. 2 ENRICHM.CHAR...
 22. 2 TABLE.COLUMN...
 23. 2 TABLE.LINE.....
 24. 2 ENRI.COL.......
 25. 2 PROMPT.COL.....
 26. 2 RESERVED05.....
 27. 2 RESERVED04.....
 28. 2 RESERVED03.....
 29. 2 RESERVED02.....
 30. 2 RESERVED01.....
 31. 2. 1 P
 32. 2. 1 TOOL.TIP....
 33. 2 DROP
 34. 2 ENQ.
 35. 2 POPUP.CONTROL..
 36. 2 CASE.CONV......
 37. 2 HYPE
 38. 2. 1 I.LINK......
 39. 2 ASSOCIATION....
 40. 2 DISPLAY.TYPE...
 41. 2 I.DESC.........
 42. 2. 1 ATTRIBS..... HOT.FIELD
```

Whenever we expand the multi-value using “>” or “<” key, the whole group is expanded.  Go to the field #42.3.1, then type > Enter and see group #4 appearing:

T24

```
Model Bank R14          VERSION, INPUT

    PGM.NAME.VERSION.. FUNDS.TRANSFER,TRAIN
--------------------------------------------------------------------------------
 37. 3 HYPE
 38. 3. 1 I.LINK......
 39. 3 ASSOCIATION....
 40. 3 DISPLAY.TYPE...
 41. 3 I.DESC.........
 42. 3. 1 ATTRIBS.....
 13. 4 FIELD.NO.......
 14. 4 COLUMN.........
 15. 4 EXPANSION......
 16. 4 TEXT.CHAR.MAX..
 17. 4. 1 TEXT........
 18. 4. 1 TXT.040..078
 19. 4. 1 TXT.079..117
 20. 4. 1 TXT.118..132
 21. 4 ENRICHM.CHAR...
 22. 4 TABLE.COLUMN...
--------------------------------------------------------------------------------
07 OCT 2014 21:21:17  USER (22 APR) INPUTTER         [22662,IPAGE 7   >>14>>>
ACTION
```

To delete the whole group type - (dash) at the field 13.4, then Enter.

Such groups are called “multi-value associations” and they are defined in the source code of application template.

Now we can test the routine.  Enter the VERSION screen, it’s different now.  As the bank day is 22 APR as indicated on the bottom part of the screen, we can try the April 30 which - since it’s the same month - can be input as just 30:

T24

```
Model Bank R14          FUNDS.TRANSFER,TRAIN INPUT           REF FT14112L7G4D


--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE..
  7 DEBIT.VALUE.DATE.. 30           Shouldn't be later than 5 work days ahead
  6 DEBIT.AMOUNT......
```

And if we input, say, 25:

T24

```
Model Bank R14          FUNDS.TRANSFER,TRAIN INPUT            REF FT14112L7G4D


--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE..
  7 DEBIT.VALUE.DATE.. 25 APR 2014
  6 DEBIT.AMOUNT...... _
```

Now let’s go to Browser:

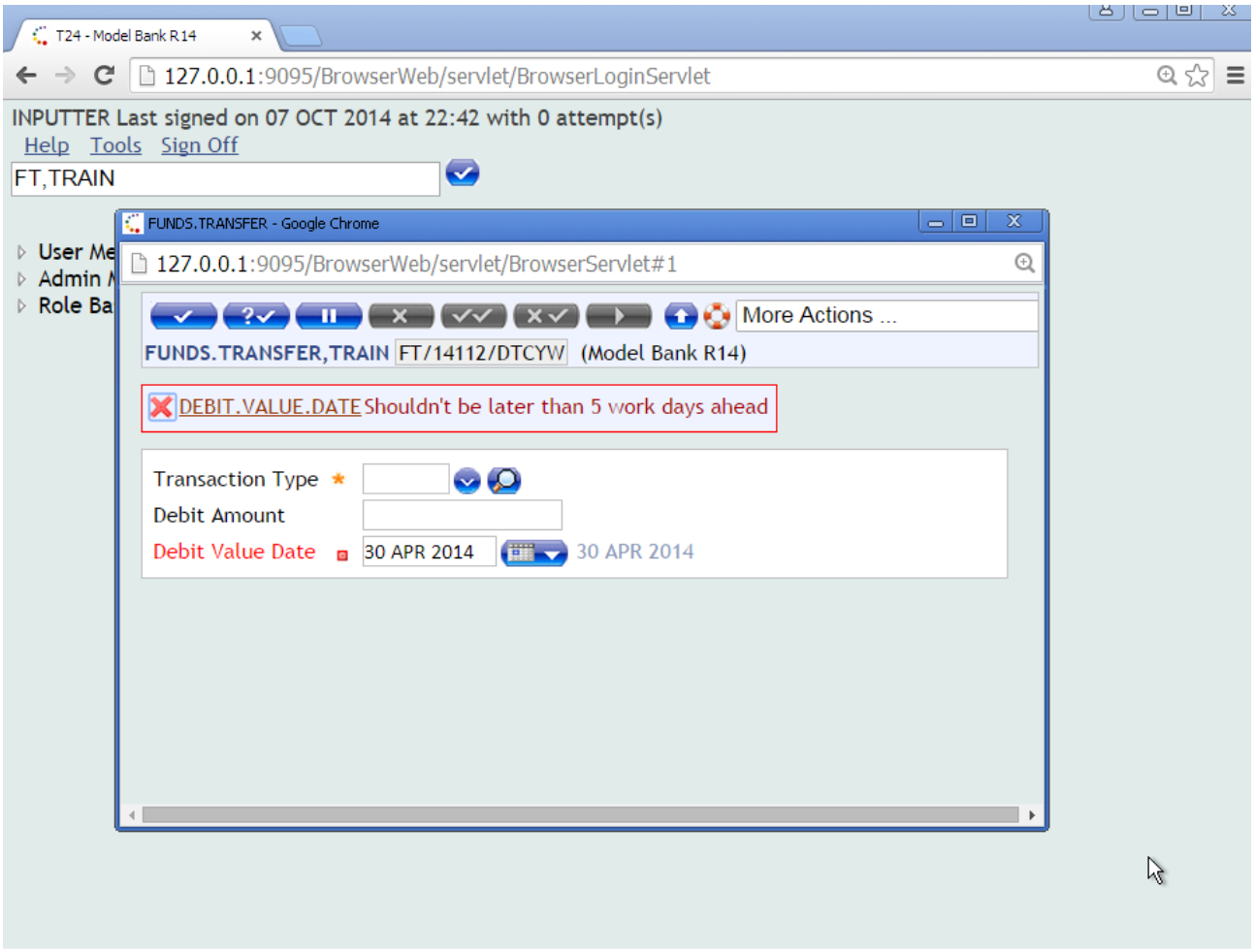


**Figure 19:  Validation error in Browser.**

*(Asterisk near the field Transaction Type means that this field is mandatory.  Besides the fields that are mandatory according to core business logic, in VERSION we can assign this attribute to any other field.)*

*(Small red rectangle near the field Debit Value Date signifies the hot field.  The data exchange with the server happens whenever this field is amended.  Having too many hot fields increases the network traffic and therefore is not recommended.)*

*(If field is not a hot field, in Browser it will be validated only when user clicks “commit” or “validate” button.)*

*(Click on a link near the error message at the top brings us directly to the field where the error occurred.)*

## 1.35 Some useful T24 applications

Where the current working day of the bank is stored? It’s not necessarily the same as the calendar date...

In the DATES table (output after DATES L):

T24

```
Model Bank R14           DATES - Default List

   ID               TODAY        VERSION  LAST.WORKING.DAY  PERIOD.END  FUNCT.
-------------------------------------------------------------------------------
1  EU0010001        22 APR 2014        1       17 APR 2014     20140422
2  EU0010001-COB    22 APR 2014        1       17 APR 2014     20140422
3  GB0010001        22 APR 2014        1       17 APR 2014     20140422
4  GB0010001-COB    22 APR 2014        1       17 APR 2014     20140422
5  GB0010002        22 APR 2014        1       17 APR 2014     20140422
6  GB0010002-COB    22 APR 2014        1       17 APR 2014     20140422
7  GB0010003        22 APR 2014        1       17 APR 2014     20140422
8  GB0010003-COB    22 APR 2014        1       17 APR 2014     20140422
9  GB0010004        22 APR 2014        1       17 APR 2014     20140422
10 GB0010004-COB    22 APR 2014        1       17 APR 2014     20140422
11 GB0010005        22 APR 2014        1       17 APR 2014     20140422
12 GB0010005-COB    22 APR 2014        1       17 APR 2014     20140422
13 SG0010001        22 APR 2014        1       17 APR 2014     20140422
14 SG0010001-COB    22 APR 2014        1       17 APR 2014     20140422


-------------------------------------------------------------------------------
07 OCT 2014 21:53:19  USER (22 APR) INPUTTER          [31433,IPAGE   1 >>>1>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

*(“-COB” records are for supporting the input to the next business day during Close of Business process when T24 “non-stop” module is installed.)*

To see the full record for the main branch type at “AWAITING APPLICATION” prompt: DATES S BNK

T24

```
Model Bank R14          DATES SEE

    COMPANY.CODE...... GB-001-0001         Model Bank R14
------------------------------------------------------------------------------
  1 TODAY.............. 22 APR 2014
  2 VERSION............ 1
  3 LAST.WORKING.DAY.. 17 APR 2014
  4 NEXT.WORKING.DAY.. 23 APR 2014
  5 LOCAL.PAYMENT.DAY. 22 APR 2014
  6 LOCAL.DISPO.DAY... 22 APR 2014
  7 BACK.VALUE.MINIMUM 26 FEB 2014
  8 BACK.VALUE.MAXIMUM 13 NOV 2013
  9 FORW.VALUE.MINIMUM 03 JUN 2014
 10 FORW.VALUE.MAXIMUM 26 AUG 2014
 11 CO.BATCH.STATUS... O
 12 PERIOD.END........ 22 APR 2014
 14 JULIAN.DATE....... 2014112
 15 CURRENT.DAY....... NORMAL
 16 LAST.PERIOD.END... 21 APR 2014
 17 BK.BAL.DATE.START. 23 JUL 1998
------------------------------------------------------------------------------
15 DEC 2014 06:07:19  USER (22 APR) INPUTTER          [22998,IPAGE 1   >>>2>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

*(We used company mnemonic - BNK - to access the record.)*

Note that @ID - COMPANY.CODE - is displayed with dashes but they don’t present in it.  This is defined in corresponding STANDARD.SELECTION record:

T24

```
Model Bank R14          STANDARD SELECTION FIELDS SEE

    FILE.NAME......... COMPANY
------------------------------------------------------------------------------
  1. 1 SYS.FIELD.NAME. @ID
  2. 1 SYS.TYPE....... D
  3. 1. 1 SYS.FIELD.NO 0
  4. 1. 1 SYS.VAL.PROG IN2COM&&&R##-###-####                  <=====
  6. 1 SYS.DISPLAY.FMT 11R
  7. 1 SYS.ALT.INDEX.. N
 10. 1 SYS.SINGLE.MULT S
 11. 1 SYS.LANG.FIELD. N
 ...
```

To get TODAY in jQL:

jsh

```
LIST F.DATES TODAY
```

How T24 knows which days are holidays or weekends?  From HOLIDAY table:

T24

```
Model Bank R14          HOLIDAY SEE

    HOLIDAY.CODE...... GB 00 2014          Great Britain
--------------------------------------------------------------------------------
  3 MARCH.............. 25 26 27
  4 APRIL.............. 04 10 15 16 18 21
 13. 1 WEEKEND.DAYS... SA
 13. 2 WEEKEND.DAYS... SU
 14 MTH.01.TABLE...... WWWHHWWWWWHHWWWWWHHWWWWWHHWWWWW
 15 MTH.02.TABLE...... HHWWWWWHHWWWWWHHWWWWWHHWWWWXXX
 16 MTH.03.TABLE...... HHWWWWWHHWWWWWHHWWWWWHHWHHHHWHW
 17 MTH.04.TABLE...... WWWHHHWWWHWHHWHHWHHHHWWWWHHWWWX
 18 MTH.05.TABLE...... WWHHWWWWWHHWWWWWHHWWWWWHHWWWWWH
 19 MTH.06.TABLE...... HWWWWWHHWWWWWHHWWWWWHHWWWWWHHWX
 20 MTH.07.TABLE...... WWWWHHWWWWWHHWWWWWHHWWWWWHHWWWW
 21 MTH.08.TABLE...... WHHWWWWWHHWWWWWHHWWWWWHHWWWWHH
 22 MTH.09.TABLE...... WWWWWHHWWWWWHHWWWWWHHWWWWWHHWWX
 23 MTH.10.TABLE...... WWWHHWWWWWHHWWWWWHHWWWWWHHWWWWW
 24 MTH.11.TABLE...... HHWWWWWHHWWWWWHHWWWWWHHWWWWHHX
 25 MTH.12.TABLE...... WWWWWHHWWWWWHHWWWWWHHWWWWWHHWWW
--------------------------------------------------------------------------------
07 OCT 2014 21:55:37  USER (22 APR) INPUTTER          [31433,IPAGE 1   >>>2>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

Each month table is a line of characters representing days from 1 to 31; “W” is a work day, “H” is weekend or holiday, “X” is a non-existing day like 31st of April.

Table for April is:  17 MTH.04.TABLE......  WWWHHHWWWHWHHWHHWHHHHWWWWHHWWWX. So, difference between 22/04/2014 and 30/04/2014 is 6 work days and it triggered the error in the recent example.

## 1.36  No-input fields

In HOLIDAY table shown in the previous chapter fields 14 - 25 are no-input.  It’s not possible to navigate to, say, field 17 typing 17 Enter - T24 won’t react.  It’s been set up in corresponding STANDARD.SELECTION record.

T24

```
Model Bank R14          STANDARD SELECTION FIELDS SEE

    FILE.NAME......... HOLIDAY
--------------------------------------------------------------------------------
 11.18 SYS.LANG.FIELD. N
 12.18 SYS.GENERATED.. Y
  1.19 SYS.FIELD.NAME. MTH.04.TABLE
  2.19 SYS.TYPE....... D
  3.19. 1 SYS.FIELD.NO 17
  4.19. 1 SYS.VAL.PROG IN2A&&NOINPUT                          <=====
  6.19 SYS.DISPLAY.FMT 31L
  7.19 SYS.ALT.INDEX.. N
 10.19 SYS.SINGLE.MULT S
 11.19 SYS.LANG.FIELD. N
 12.19 SYS.GENERATED.. Y
  1.20 SYS.FIELD.NAME. MTH.05.TABLE
  2.20 SYS.TYPE....... D
  3.20. 1 SYS.FIELD.NO 18
  4.20. 1 SYS.VAL.PROG IN2A&&NOINPUT
  6.20 SYS.DISPLAY.FMT 31L
--------------------------------------------------------------------------------
07 OCT 2014 22:14:11  USER (22 APR) INPUTTER          [19579,IPAGE 11  >>28>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

*(And STANDARD.SELECTION is built from the data that is set in the source code of corresponding application template.)*

Dictionary entry for an application field doesn’t contain anything regarding no-input attribute or like (e.g.  NOCHANGE); all such checks are done not on DBMS level but in T24 core code:

jsh

```
CT DICT F.HOLIDAY 'MTH.04.TABLE'
```

Output

```
    MTH.04.TABLE
001 D
002 17
003
004 MTH.04.TABLE
005 31L
006 S
007
...
030 JBASE_EDICT_START
031 108
032
033
034 MTH_04_TABLE
035
036
```

```
037
038
039 1073741824
040
041
042
043
044
045
046
047 JBASE_EDICT_END
```

## 1.37 VERSION - cross-validation

“Input” routine is the VERSION hook that provides this functionality.

First of all, not all hook routines are triggered in some particular case - for example, “input” and “validation” routines aren’t triggered when user puts record on HOLD.

“AUT.NEW.CONTENT” routine won’t run on “S” function.

Other rules - depending on application business logic - may apply.

To be able to test “input” routine we need all core checks for the record to be done successfully and only after that local routines will run.

As FT is quite complex application and requires certain business knowledge, let’s try to see if there are any unauthorised records where some insignificant fields can be amended and then upon commit our routine would be triggered.

Thing to test will be if debit currency and credit currency are the same.  To have them different is generally a normal thing but let’s assume that for our VERSION such records are inappropriate.

jBC

```
SUBROUTINE SUBR.INP

$INSERT I_COMMON
$INSERT I_EQUATE

   dr_ccy = R.NEW(FundsTransfer_DebitCurrency)
   cr_ccy = R.NEW(FundsTransfer_CreditCurrency)

   IF dr_ccy NE cr_ccy THEN
      AF = FundsTransfer_DebitCurrency      ;* set where the error will be displayed
      ETEXT = "Doesn't match credit currency"
      CALL STORE.END.ERROR

      AF = FundsTransfer_CreditCurrency     ;* another error
      ETEXT = "Doesn't match debit currency"
      CALL STORE.END.ERROR
   END
   RETURN
END
```

Compile it....  oops...

jsh

```
Record 'SUBR.INP' written to file 'ETC.BP'

jsh r14  -->BASIC -I../T24_BP ETC.BP SUBR.INP
SUBR.INP
Warning: Variable FundsTransfer_DebitCurrency is never assigned!
Warning: Variable FundsTransfer_CreditCurrency is never assigned!
BASIC_8.c
Source file SUBR.INP compiled successfully
jsh r14  -->
```

We forgot the FT insert file.  Add the following line after “$INSERT I_EQUATE”:

jBC

```
$INSERT I_F.FUNDS.TRANSFER
```

Then we need to add these 2 fields to our VERSION design:

T24

```
 Model Bank R14          VERSION, SEE

     PGM.NAME.VERSION.. FUNDS.TRANSFER,TRAIN
 ------------------------------------------------------------------------------
   2 RECORDS.PER.PAGE.. 1
   3 FIELDS.PER.LINE... 1
  13. 1 FIELD.NO....... TRANSACTION.TYPE    TRANSACTION.TYPE
  13. 2 FIELD.NO....... DEBIT.VALUE.DATE    DEBIT.VALUE.DATE
  42. 2. 1 ATTRIBS..... HOT.FIELD
  13. 3 FIELD.NO....... DEBIT.AMOUNT        DEBIT.AMOUNT
  13. 4 FIELD.NO....... DEBIT.CURRENCY      DEBIT.CURRENCY
  13. 5 FIELD.NO....... CREDIT.CURRENCY     CREDIT.CURRENCY
  46 NO.OF.AUTH........ 1
  50. 1 AUTOM.FIELD.NO. DEBIT.THEIR.REF     DEBIT.THEIR.REF
  52. 1 AUT.NEW.CONTENT @SUBR.ANC
  55. 1. 1 VAL.ASSOC... COMMISSION.TYPE
  55. 1. 2 VAL.ASSOC... COMMISSION.FOR
  55. 2. 1 VAL.ASSOC... CHARGE.TYPE
  55. 2. 2 VAL.ASSOC... CHARGE.FOR
  55. 3. 1 VAL.ASSOC... TAX.TYPE
 ------------------------------------------------------------------------------
 08 OCT 2014 00:13:28  USER (22 APR) INPUTTER          [29894, PAGE 1   >>>3>>>
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

*(VAL.ASSOC noinput field holds information about multi-value associations derived from application template source code.)*

Also - create EB.API record for out “input” routine (same procedure as for ”validation” one earlier) and attach this routine to our VERSION:

T24

```
Model Bank R14          VERSION, INPUT

    PGM.NAME.VERSION.. FUNDS.TRANSFER,TRAIN
-------------------------------------------------------------------------------
 62 D.SLIP.TRIGGER....
 63. 1 INPUT.ROUTINE.. SUBR.INP                                    <=====
 64. 1 AUTH.ROUTINE ..
 65 REPORT.LOCKS...... YES
 66 GTS.CONTROL.......
 67. 1 GB D
 68. 1 ASSOC.VERSION..
 69 NEXT.VE
 70 INITIAL.CURSOR.POS
 71. 1 RESERVED.4.....
 72. 1 CAPTION........
 73 EXC.INC.RTN.......
 74. 1 ID.RTN.........
 75. 1 CHECK.REC.RTN..
 76. 1 AFTER.UNAU.RTN.
 77. 1 BEFORE.AUTH.RTN
-------------------------------------------------------------------------------
08 OCT 2014 00:27:14  USER (22 APR) INPUTTER          [17518,IPAGE 13  >>16>>>
ACTION
```

*(Field 63 is multi-valued so it’s possible to attach more than one routine there.)*

To find the record that will trigger the error we’ll use the search (which is impossible in Browser - it can’t compare 2 fields, only a field with a constant or a pattern):

jsh

```
LIST FBNK.FUNDS.TRANSFER$NAU WITH DEBIT.CURRENCY NE CREDIT.CURRENCY
```

Output

```
@ID.....................    FT1.

FT141126TL15                OT40
FT14112H3HM6                OT12
FT14112QRRX3                AC
FT14097TZBNX                OT03
FT14097YKQLL                AC
FT141126GVXY                AC
...
```

Output is a bit unexpected since there are many fields in FT...  Actually, not all available fields present in the output of LIST command but only those which are mentioned in dictionary item “@”:

jsh

```
CT DICT FBNK.FUNDS.TRANSFER @
```

Output

```
    @
001 PH
002 FT1
```

We listed CUSTOMER earlier, what’s in its dictionary?

jsh

```
CT DICT FBNK.CUSTOMER @
```

Output

```
    @
001 PH
002 @ID CUSTOMER.CODE MNEMONIC SHORT.NAME NAME.1 NAME.2 STREET TOWN.COUNTRY RE
    LATION.CODE REL.CUSTOMER REVERS.REL.CODE SECTOR ACCOUNT.OFFICER OTHER.OFFI
    CER INDUSTRY TARGET NATIONALITY CUSTOMER.STATUS RESIDENCE CONTACT.DATE INT
    RODUCER TEXT LEGAL.ID REVIEW.FREQUENCY BIRTH.INCORP.DATE GLOBAL.CUSTOMER C
    USTOMER.LIABILITY LANGUAGE LOCAL.REF OVERRIDE RECORD.STATUS CURR.NO INPUTT
    ER DATE.TIME AUTHORISER CO.CODE DEPT.CODE AUDITOR.CODE AUDIT.DATE.TIME POS
    TING.RESTRICT DISPO.OFFICER POST.CODE COUNTRY CONFID.TXT DISPO.EXEMPT ISSU
    E.CHEQUES CLS.CPARTY FX.COMM.GROUP.ID RESIDENCE.REGION COMPANY.BOOK ASSET.
    CLASS CUSTOMER.RATING ADDRESS CR.PROFILE.TYPE CR.PROFILE NO.UPDATE.CRM TIT
...
```

Back to choosing an FT record.  Double-check the record contents adding field names to command:

jsh

```
LIST FBNK.FUNDS.TRANSFER$NAU DEBIT.CURRENCY CREDIT.CURRENCY
   WITH DEBIT.CURRENCY NE CREDIT.CURRENCY
```

(*Again, all jQL queries are one-liners!*)

Output

```
@ID.....................    DEBIT.CURRENCY    CREDIT.CURRENCY

FT141126TL15                USD               GBP
FT14112H3HM6                USD               CHF
FT141126GVXY                EUR               GBP
FT14093171NR                USD               GBP
...
```

Try to validate the deal FT141126TL15.  Inputter was different hence the warning:

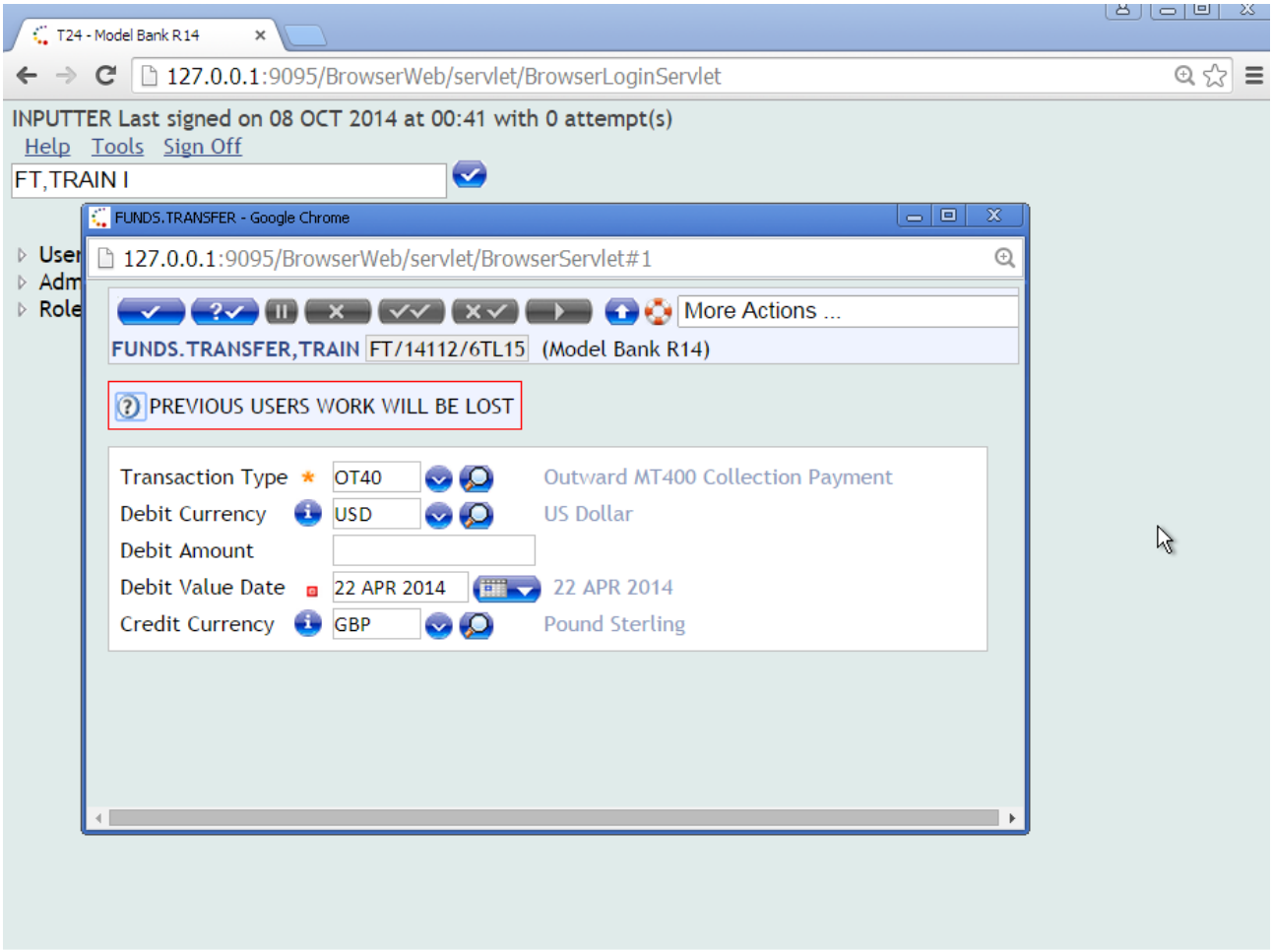


**Figure 20:  Validate FT deal screen 1.**

Click “validate”:

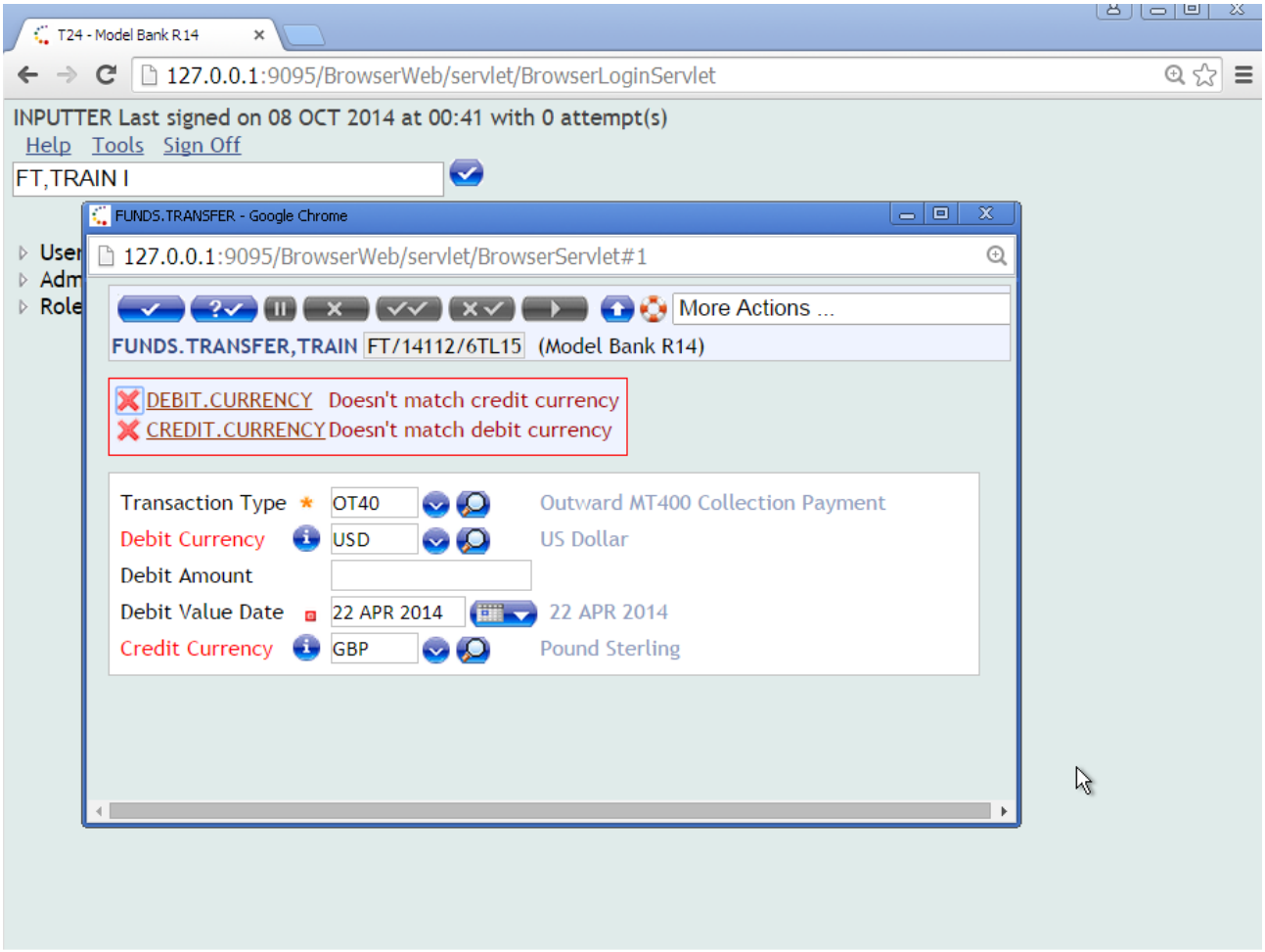


**Figure 21: Validate FT deal screen 2.**

Earlier we saw that we can input a date in the current month just typing the day number.  Other shortcuts exist in T24, e.g.  for amount we can type 1T for one thousand, 1M for one million, 1B for one billion:

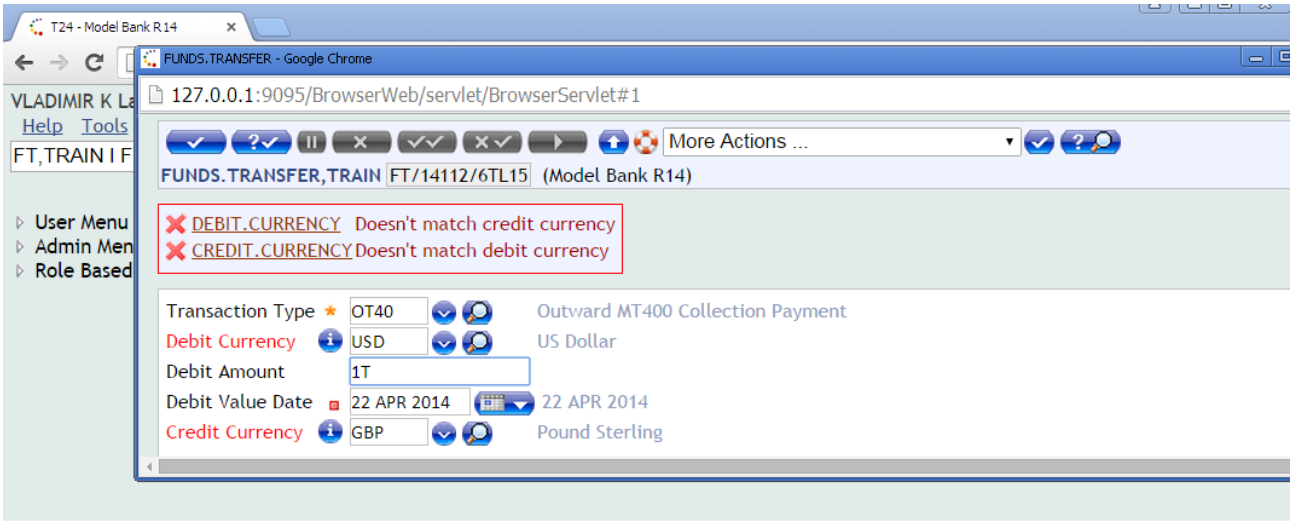


**Figure 22:  Shortcut for amount input.**

After “validate” button is clicked, the value 1000.00 appears in “Debit Amount” field.

These shortcuts work in Classic interface as well.

## 1.38  VERSION - other hooks

Names are self-explanatory; more information and examples later:

T24

```
 Model Bank R14          VERSION INPUT

     PGM.NAME.VERSION.. FUNDS.TRANSFER,QWERTY
 ------------------------------------------------------------------------------
...
  64. 1 AUTH.ROUTINE ..
...
  74. 1 ID.RTN.........
  75. 1 CHECK.REC.RTN..
  76. 1 AFTER.UNAU.RTN.
  77. 1 BEFORE.AUTH.RTN
...
 ------------------------------------------------------------------------------
```

## 1.39 CDD fatal error

The logic of validation routine SUBR.VAL, however, is not fully correct. If we try to validate a deal with empty DEBIT.VALUE.DATE in Browser, the session will “hang”. Open a record:

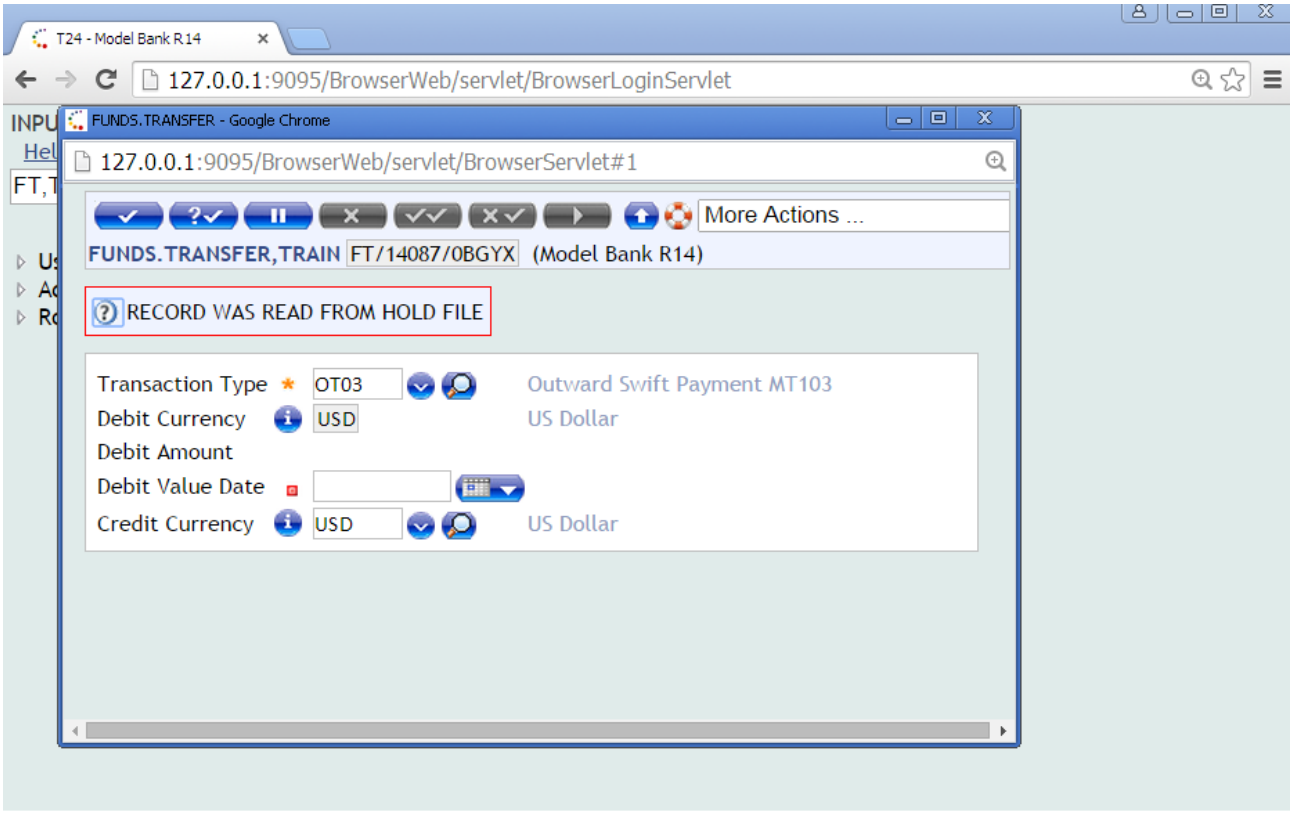


**Figure 23: Validate FT deal - fatal error expected.**

Click “validate”; after some time the following screen appears:

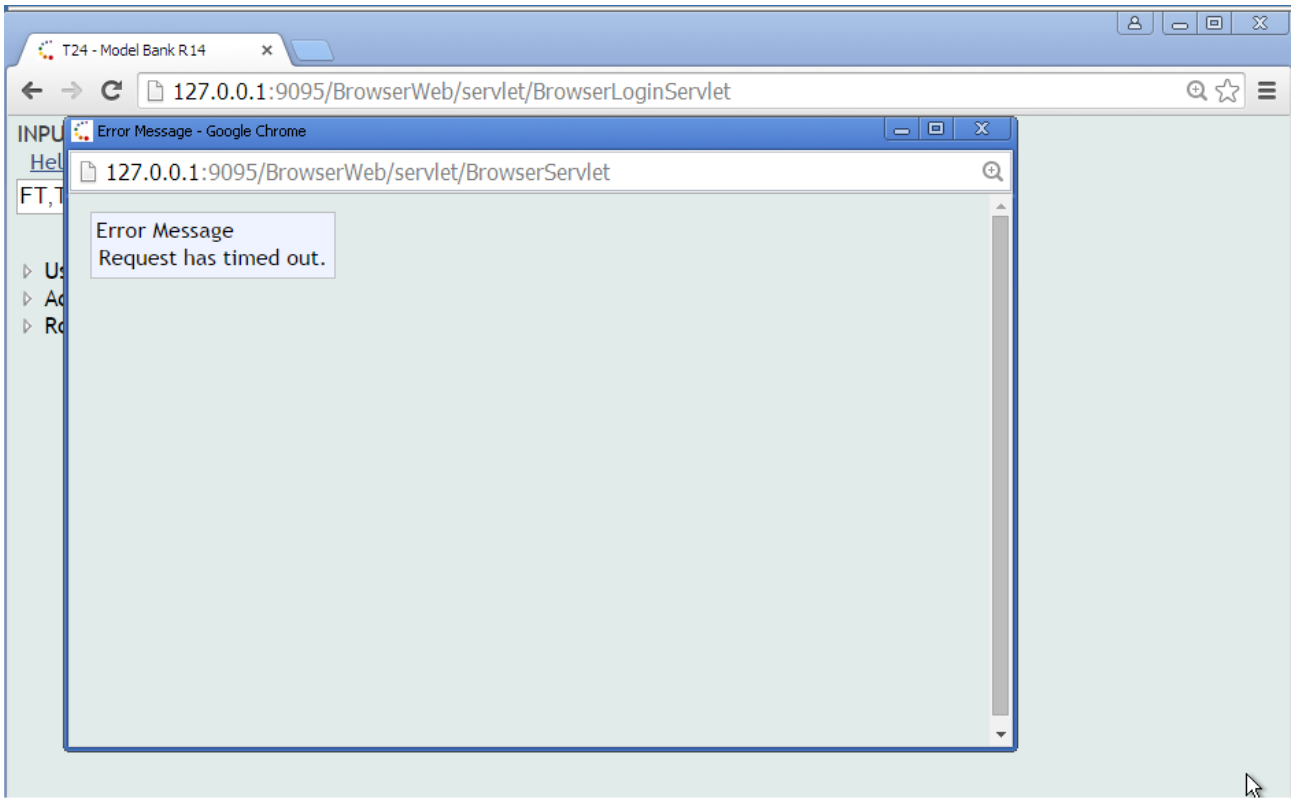


**Figure 24:  Validate FT deal - fatal error.**

To understand why, try to commit this record in Classic mode:

T24

```
 Model Bank R14          FUNDS.TRANSFER,TRAIN INPUT            REF FT140870BGYX


 ------------------------------------------------------------------------------
   1 TRANSACTION.TYPE.. OT03                Outward Swift Payment MT103
   7 DEBIT.VALUE.DATE..
   6 DEBIT.AMOUNT......
   5 DEBIT.CURRENCY.... USD                 US Dollar
  13 CREDIT.CURRENCY... USD                 US Dollar

...


 ------------------------------------------------------------------------------
 08 OCT 2014 01:14:21  USER (22 APR) INPUTTER          [29555,I      VALIDATED
 ACTION

** FATAL ERROR IN (CDD) **
DATE WITHOUT 8 CHARACTERS

jsh r14  -->
```

This error isn’t visible to end user in Browser but it can be found in JBoss log (use JED in readonly mode in order not to lock the file):

jsh

```
JED D:\Temenos\Modelbank-R14\Infra\Jboss\server\default\log server.log (R
                                                                       ^^^
```

JED

```
READ ONLY File D:\Temenos\Modelbank-R14\Infra\Jboss        default\log 06:38:20
Command->
0001 2014-12-15 09:50:09,046 INFO  [org.jboss.web.WebService] (main) Using RMI
0002 2014-12-15 09:50:58,875 INFO  [org.jboss.wsf.stack.jbws.NativeServerConfig
0003 2014-12-15 09:50:58,875 INFO  [org.jboss.wsf.stack.jbws.NativeServerConfig
0004 2014-12-15 09:51:01,281 INFO  [org.jboss.dependency.plugins.AttributeCallb
0005 2014-12-15 09:51:06,234 INFO  [org.jboss.logbridge.LogNotificationListener
0006 2014-12-15 09:52:07,781 INFO  [org.jboss.ejb3.deployers.Ejb3DependenciesDe
0007 2014-12-15 09:52:07,781 INFO  [org.jboss.ejb3.deployers.Ejb3DependenciesDe
...
```

Search “CDD”: Press Esc to go to command line, then enter command /CDD

JED

```
String 'CDD' found                                  +100  Insert      01:37:09
Command->
01175 nnectionFactoryBrowser] (http-0.0.0.0-9095-5) ** FATAL ERROR IN (CDD) **
01176 nnectionFactoryBrowser] (http-0.0.0.0-9095-5) DATE WITHOUT 8 CHARACTERS
01177 0.0-9095-5) T24 transaction failed [5624416b-67cc-45ac-a462-ca14e2711359]
01178 pting to re-send the request [1. retry] due to connection error: T24 tran
01179 ng request to server
01180  Failed to receive message
01181 .0-9095-5) Connection error occured: org.jboss.resource.connectionmanager
```

To correct this error we need to add the following check - if field content consists of 8 numeric characters:

jBC

```
SUBROUTINE SUBR.VAL

$INSERT I_COMMON
$INSERT I_EQUATE

   IF NOT(COMI MATCHES "8N") THEN RETURN          ;* otherwise CDD fails

   dr_val_date = COMI                             ;* global var, contains user input
   the_diff = 'W'                          ;* 'W' for work days, 'C' for calendar days
   CALL CDD('', TODAY, dr_val_date, the_diff)        ;* TODAY is another global var;
                                                     ;* never amend it!
   IF the_diff GT 5 THEN ETEXT = "Shouldn't be later than 5 work days ahead"

   RETURN
END
```

Now see the screen after “validate” button was clicked:

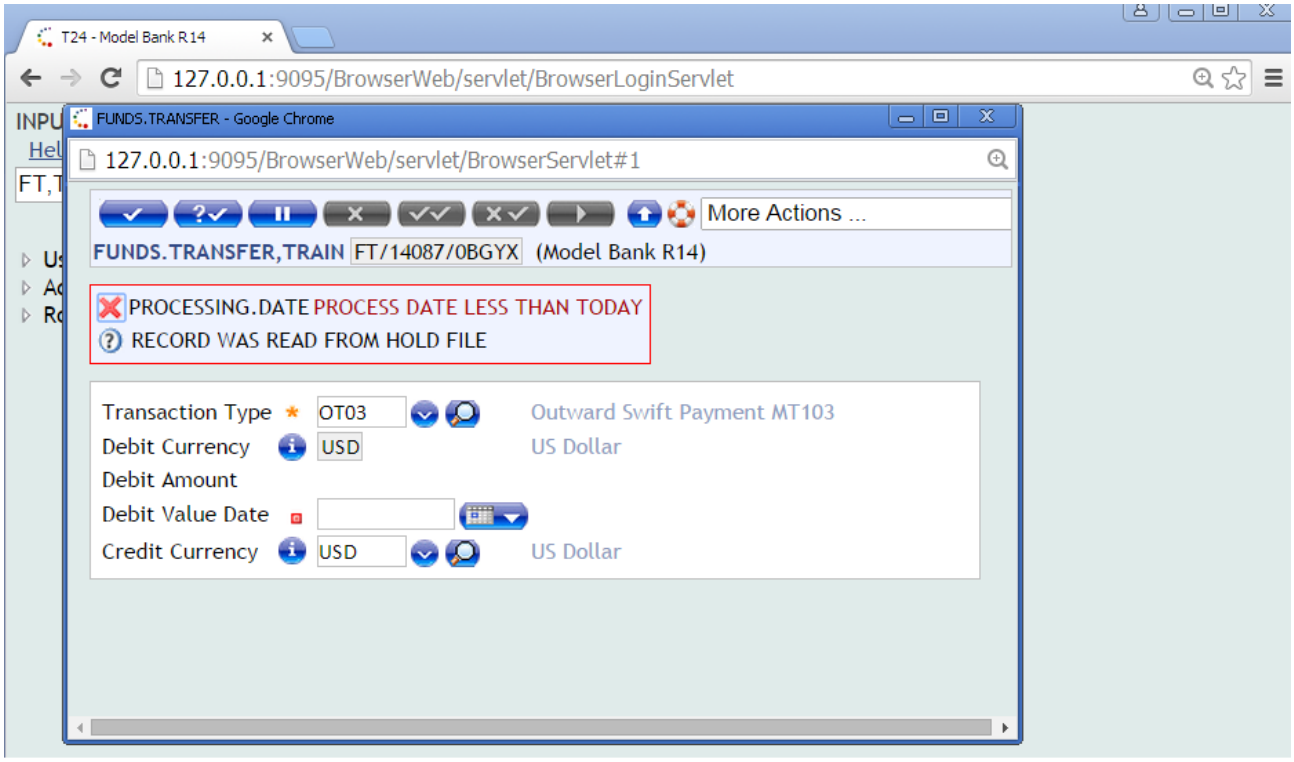


**Figure 25:  Validate FT deal - fatal error avoided.**

Now we can see that local field level validations passed correctly and what we see now is core cross-validation that goes after it but before local cross-validation.

Core field-level checks come before local ones.  In the following screen user typed 12345678 to the date field and clicked “validate”:

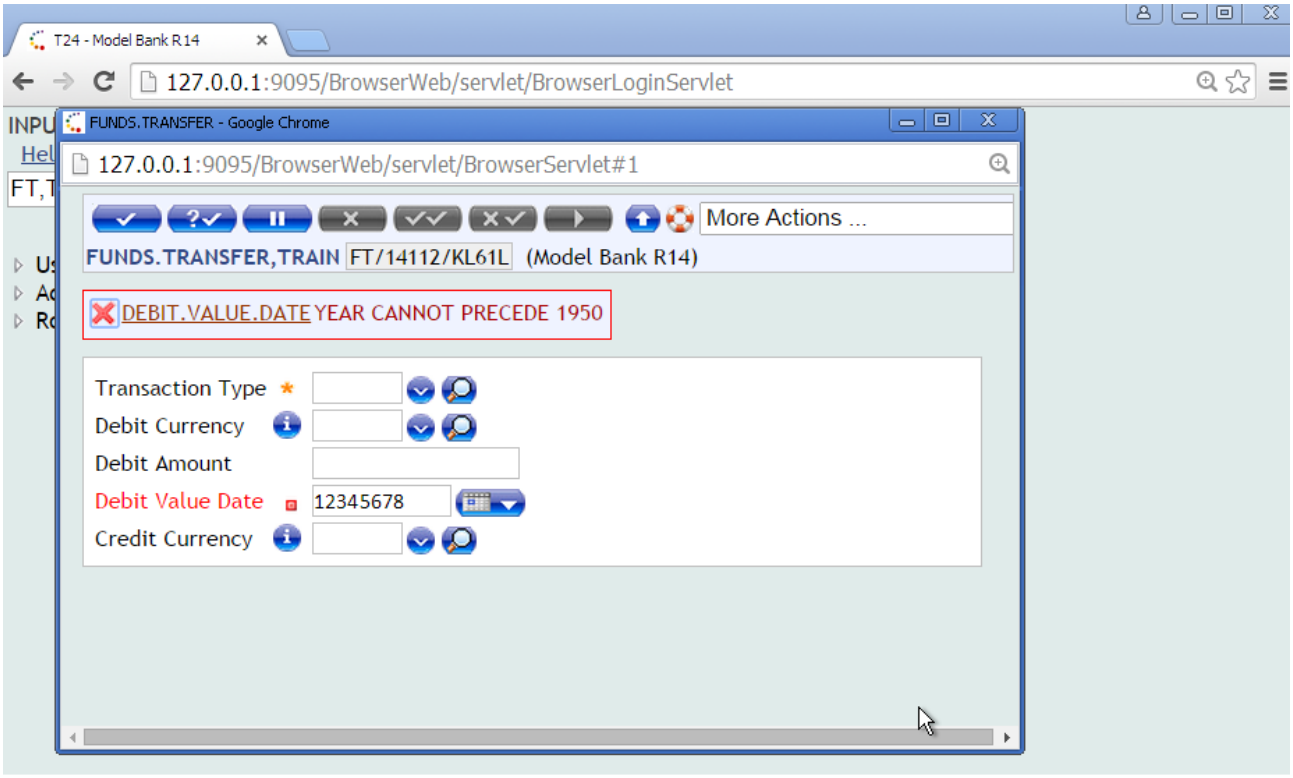


**Figure 26:  Validate FT deal - core field-level check.**

*(For year to preceed 1950, T24 data type should be “DD” rather than “D”. But then the next error would be risen regarding month number being 56.)*

## 1.40  OFS

OFS stands for “open financial service”.  Messages (in text or data files) in OFS syntax can be used to input, verify or extract data from T24.  OFS syntax is used in Browser communications and it’s the only valid way to input T24 data (except manual input in T24 via terminal mode).

OFS can be tested from jsh prompt using tSS program.  As parameter, tSS takes @ID if OFS.SOURCE (T24 application where OFS parameters are listed).

We'll take the record TAABS which has minimum settings required for fuctioning:

T24

```
Model Bank R14          OFS SOURCE SEE

    SOURCE.NAME....... TAABS
------------------------------------------------------------------------------
  1 DESCRIPTION....... FOR TAG
  2 SOURCE.TYPE....... TELNET
  3. 1 LOGIN.ID....... ANY
 14 LOG.DETAIL.LEVEL.. NONE
 24 SYNTAX.TYPE....... OFS
 26 GENERIC.USER...... INPUTTER
 42 CURR.NO........... 1
 43. 1 INPUTTER....... 15012_VLADIMIR.K
 44. 1 DATE.TIME...... 05 OCT 14 11:56
 45 AUTHORISER........ 15012_VLADIMIR.K
 46 CO.CODE........... GB-001-0001         Model Bank R14
 47 DEPT.CODE......... 1                   Implementation




------------------------------------------------------------------------------
08 OCT 2014 02:09:22  USER (22 APR) INPUTTER          [3802,INPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

So:

jsh

```
tSS TAABS
```

Output

```
<tSS version="1.1"><t24version>R14</t24version><t24pid>2508</t24pid><t24ofssource>
TAABS</t24ofssource><clientIP/></tSS>
```

Message structure for data input or validation:

OFS message format

```
VERSION/Function/Mode/parameter 1/parameter 2/etc,T24 login/password[/branch],@ID,
field name:[value]:[subvalue]=contents,
field name 2:[value]:[subvalue]=contents,
other fields follow...
```

*(OFS message is one-liner as well as jQL query.)*

Same VERSIONs can be used both for manual input and OFS (though display of fields makes no sence for the latter).

Validate a deal using our FT VERSION:

OFS input

```
FT,TRAIN/I/VALIDATE,INPUTT/12346,FT140870BGYX,
```

OFS output

```
SECURITY VIOLATION DURING SIGN ON PROCESS
```

Correct password...

OFS input

```
FT,TRAIN/I/VALIDATE,INPUTT/123456,FT140870BGYX,
```

Output is the same as shown at the Figure 25:

OFS output

```
FT140870BGYX//-1/NO,PROCESSING.DATE:1:1=PROCESS DATE LESS THAN TODAY
```

Try to correct the situation by supplying some appropriate value:

OFS input

```
FT,TRAIN/I/VALIDATE,INPUTT/123456,FT140870BGYX,PROCESSING.DATE::=20140422
```

In case of success output contains the whole record and is quite long; see the beginning and the end:

OFS output

```
<requests><request>FT140870BGYX//1,TRANSACTION.TYPE:1:1=OT03,DEBIT.ACCT.NO:1:1=70467,
CURRENCY.MKT.DR:1:1=1,DEBIT.CURRENCY:1:1=USD,DEBIT.VALUE.DATE:1:1=20140422,
DEBIT.THEIR.REF:1:1=TRAIN DEAL,CREDIT.ACCT.NO:1:1=23884,,CURRENCY.MKT.CR:1:1=1,
CREDIT.CURRENCY:1:1=USD,CREDIT.AMOUNT:1:1=3000.00,CREDIT.VALUE.DATE:1:1=20140422,
...
RECORD.STATUS:1:1=INAU,CURR.NO:1:1=1,INPUTTER:1:1=26746_INPUTTER__OFS_TAABS_
FT140870BGYX,DATE.TIME:1:1=1410080230,CO.CODE:1:1=GB0010001,DEPT.CODE:1:1=1
</request></requests>
```

Field of interest:

RECORD.STATUS:1:1=INAU. We have number of authorisations = 1 in our VERSION. In order to create authorised record we have several options:

- Post another OFS message with “A” function.
- Use zero-auth VERSION from the beginning.
- Redefine number of authorisations in OFS message.

See the last method applied (so far we're validating so no real commit happens):

OFS input

```
FT,TRAIN/I/VALIDATE//0,INPUTT/123456,FT140870BGYX,PROCESSING.DATE::=20140422
                   ^^^
```

OFS output

```
...
INPUTTER:1:1=26746_INPUTTER__OFS_TAABS_FT140870BGYX,DATE.TIME:1:1=1410080238,
AUTHORISER:1:1=26746_INPUTTER_OFS_TAABS_FT140870BGYX,CO.CODE:1:1=GB0010001,
DEPT.CODE:1:1=1</request></requests>
```

Now we see AUTHORISER field in OFS output.

To proceed with real input substitute "/VALIDATE/" to "/PROCESS/" in OFS message. Let's create a record "SEESPF" in the simplest T24 application - ABBREVIATION - using "comma" VERSION:

OFS input

```
ABBREVIATION,/I/PROCESS,INPUTT/123456,SEESPF,ORIGINAL.TEXT::="SPF S SYSTEM"
```

OFS output

```
SEESPF//1,ORIGINAL.TEXT:1:1=SPF S SYSTEM,RECORD.STATUS:1:1=INAU,CURR.NO:1:1=1,
INPUTTER:1:1=13212_INPUTTER__OFS_TAABS,DATE.TIME:1:1=1410080244,CO.CODE:1:1=
GB0010001,DEPT.CODE:1:1=1
```

Why INAU? Check the VERSION:

jsh

```
LIST F.VERSION 'ABBREVIATION,' NO.OF.AUTH
```

Output

```
@ID.............................................    NO.OF.AUTH


 No Records Listed
```

So there's no VERSION. Even if we created one with NO.OF.AUTH = 0, OFS wouldn't concede to it - that's the exception in the rule.  The easiest way is to use "//0" after "/PROCESS":

OFS input

```
ABBREVIATION,/I/PROCESS//0,INPUTT/123456,SEESPF,ORIGINAL.TEXT::="SPF S SYSTEM"
```

OFS output

```
SEESPF//-1/NO,NOT AUTH. RECORD NOT CHANGED
```

OK, let’s force it forward.  Slightly amend the record, then return the necessary value back:

OFS input

```
ABBREVIATION,/I/PROCESS//0,INPUTT/123456,SEESPF,ORIGINAL.TEXT::="SPF I SYSTEM"
ABBREVIATION,/I/PROCESS//0,INPUTT/123456,SEESPF,ORIGINAL.TEXT::="SPF S SYSTEM"
```

Now we can go to T24, type SEESPF at “AWAITING APPLICATION” prompt and see SPF record:

T24

```
 Model Bank R14          SPF SEE

     SYSTEM SPEC....... SYSTEM
 ------------------------------------------------------------------------------
   1 RUN.DATE.......... 28 MAR 2014
   2 SITE.NAME......... Model Bank R14
   3 OP.MODE........... O
   4. 1 OP.CONSOLE..... OFF
   5. 1 MAIN.ACCOUNT... ../bnk.data
   8 CURRENT.RELEASE... R14
   9 HIST.LIFE......... 1
  11 CACHE.EXPIRY...... 0
  12 ENQ.PAGE.LIMIT.... 200
  14 SYS.BACKUP.MODE... TAPE
  15 HOLD.BATCH.OUTPUT. Y
  16 MICROFICHE.OUTPUT. N
  17 REPORT.RETENTION.. 5
  19 DATA.ACC.NAME..... ../bnk.data
  20 RUN.ACC.NAME...... ../bnk.run
  21 DICT.ACC.NAME..... ../bnk.dict
 ------------------------------------------------------------------------------
 08 OCT 2014 02:51:08  USER (22 APR) INPUTTER          [27335,IPAGE 1   >>>9>>>
 ACTION _
 AWAITING PAGE INSTRUCTIONS
```

Double quotes in ORIGINAL.TEXT::="SPF I SYSTEM" were removed before field was populated.  If we need them in field contents, we can use pipe as a substitution character:

OFS input

```
ABBREVIATION,/I/VALIDATE//0,INPUTT/123456,SEESPF,ORIGINAL.TEXT::=SPF S |SYSTEM|
```

OFS output

```
SEESPF//1,ORIGINAL.TEXT:1:1=SPF S "SYSTEM",CURR.NO:1:1=2, ...
```

Comma (e.g.  in VERSION @ID) has to be substituted with a question mark:

OFS input

```
VERSION,/I/VALIDATE//0,INPUTT/123456,FUNDS.TRANSFER?TRAIN,
```

OFS output

```
FUNDS.TRANSFER,TRAIN//1,RECORDS.PER.PAGE:1:1=1,FIELDS.PER.LINE:1:1=1, ...
```

If comma is in field contents, enclose the field value in double quotes:

OFS input

```
ABBREVIATION,/I/VALIDATE//0,INPUTT/123456,SEESPF,ORIGINAL.TEXT::="SPF, I SYSTEM"
```

OFS output

```
SEESPF//1,ORIGINAL.TEXT:1:1=SPF, I SYSTEM,CURR.NO:1:1=2, ...
```

Underscore is to be enclosed in single quotes:

OFS input

```
ABBREVIATION,/I/VALIDATE//0,INPUTT/123456,SEESPF,ORIGINAL.TEXT::=SPF'_'S'_'SYSTEM
```

OFS output

```
SEESPF//-1/NO,ORIGINAL.TEXT:1:1=WRONG ALPHANUMERIC CHAR.
```

Underscore isn’t allowed in this field...  Try something else:

OFS input

```
VERSION,/I/VALIDATE//0,INPUTT/123456,FUNDS.TRANSFER?TRAIN,TEXT:1:1=Txn'_'type
```

OFS output

```
FUNDS.TRANSFER,TRAIN//-1/NO,TEXT:1:1=WRONG ALPHANUMERIC CHAR.
```

Nope.  Something else:

OFS input

```
CONTEXT.ENQUIRY,/I/VALIDATE,INPUTT/123456,ACCOUNT,ENQ.DESC:1:1=Summary'_'list
```

OFS output

```
ACCOUNT//1,DESCRIPTION:1:1=Account Enquiries,DESCRIPTION:2:1=Listes sur
les comptes,ENQUIRY.NAME:1:1=ACCT.STMT.HIST,ENQUIRY.NAME:2:1=ACCT.BAL.T
ODAY,ENQUIRY.NAME:3:1=NOSTRO.FWD.BAL,ENQUIRY.NAME:4:1=STMT.ENT.TODAY,SE
L.FIELD:1:1=STMT.ACCOUNT.NO,SEL.FIELD:2:1=ACCOUNT.NUMBER,SEL.FIELD:3:1=
ACCOUNT.ID,SEL.FIELD:4:1=ACCT.ID,CONTEXT.TYPE:1:1=ENQUIRY,CONTEXT.TYPE:
2:1=ENQUIRY,CONTEXT.TYPE:3:1=ENQUIRY,CONTEXT.TYPE:4:1=ENQUIRY,ENQ.DESC:
1:1=Summary_list,                                                  <=== here it is
ENQ.DESC:1:2=Liste des relevés,ENQ.DESC:2:1=Current balance information
,ENQ.DESC:2:2=Solde actuel,ENQ.DESC:3:1=Forward movements,ENQ.DESC:3:2=
Mouvements futurs,ENQ.DESC:4:1=Entries posted today,ENQ.DESC:4:2=Ecritu
res du jour,CURR.NO:1:1=3,INPUTTER:1:1=40608_INPUTTER__OFS_MB.LANG,DATE
.TIME:1:1=1406021620,AUTHORISER:1:1=40608_INPUTTER_OFS_MB.LANG,CO.CODE:
1:1=GB0010001,DEPT.CODE:1:1=1
```

OFS.SOURCE record BROWSERTC is used for Browser connections:

T24

```
 Model Bank R14          OFS SOURCE SEE

     SOURCE.NAME....... BROWSERTC
 ------------------------------------------------------------------------------
   1 DESCRIPTION....... FOR BROWSER CONNECTOR
   2 SOURCE.TYPE....... SESSION
  13 LOG.FILE.DIR...... BROWSERLOG
  14 LOG.DETAIL.LEVEL.. EXCEPT
  24 SYNTAX.TYPE....... OFS
  26 GENERIC.USER...... INPUTTER
  40. 1 OVERRIDE....... GB0010001
  42 CURR.NO........... 4
  43. 1 INPUTTER....... 1_DIM
  44. 1 DATE.TIME...... 28 MAR 14 10:21
  45 AUTHORISER........ 9028_INPUTTER
  46 CO.CODE........... GB-001-0001         Model Bank R14
  47 DEPT.CODE......... 1                   Implementation



 ------------------------------------------------------------------------------
 15 DEC 2014 13:25:11  USER (22 APR) INPUTTER          [6835,INPAGE 1
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

Its name is defined in JBoss settings:

jsh

```
JED D:\Temenos\Modelbank-R14\Infra\Jboss\server\default\deploy t24-ds.xml (R
```

Search:  Esc /OFS Enter

JED

```
READ ONLY File D:\Temenos\Modelbank-R14\Infra\Jboss  +2860 default\depl13:27:43
Command->
0001 Variables" type="java.lang.String">OFS_SOURCE=BROWSERTC</config-property>.
------------------------------- End Of Record --------------------------------
```

It looks like there’s no empty line at the end of the file - JED then shows all in one line.  We can add it using JED without readonly option:

jsh

```
JED D:\Temenos\Modelbank-R14\Infra\Jboss\server\default\deploy t24-ds.xml
```

End Enter Esc FI Enter

Open file again:

JED

```
File D:\Temenos\Modelbank-R14\Infra\Jboss\server\de       Insert Recor16:00:19
Command->
0001 <?xml version="1.0" encoding="UTF-8"?>
0002
0003 <!-- ====================================================================
0004 <!--
0005 <!--  JBoss Server Configuration
0006 <!--
0007 <!--  jRemote resource adapter deployment for JBoss, configured for local
0008 <!--
0009 <!-- ====================================================================
0010
0011 <connection-factories>
0012
0013 <!-- ======================= T24 Browser (20100 - 20199) ================
0014 <tx-connection-factory>
0015 .  <jndi-name>jca/t24ConnectionFactoryBrowser</jndi-name>
0016       <rar-name>tocfT24ra-ra.rar</rar-name>
0017       <connection-definition>com.temenos.tocf.t24ra.T24ConnectionFactory</
0018       <config-property name="hosts" type="java.lang.String">127.0.0.1</con
0019       <config-property name="ports" type="java.lang.String">20014</config-
0020       <config-property name="loadBalancing" type="java.lang.Boolean">true<
0021       <config-property name="allowInput" type="java.lang.Boolean">false</c
0022       <config-property name="compression" type="java.lang.Boolean">true</c
```

Go to the bottom to see the extra line that we added (Esc then line number; type 999999 to surely exceed the last line number):

JED

```
File D:\Temenos\Modelbank-R14\Infra\Jboss\server\de       Insert Recor16:01:53
Command->
0432       <config-property name="hosts" type="java.lang.String">127.0.0.1</con
0433       <config-property name="ports" type="java.lang.String">20001</config-
0434       <config-property name="loadBalancing" type="java.lang.Boolean">true<
0435       <config-property name="allowInput" type="java.lang.Boolean">false</c
0436       <config-property name="compression" type="java.lang.Boolean">true</c
0437       <config-property name="compressionThreshold" type="java.lang.Integer
0438       <config-property name="envVariables" type="java.lang.String">OFS_SOU
0439       <config-property name="actionTimeout" type="java.lang.Integer">120</
0440       <config-property name="charset" type="java.lang.String">UTF-8</confi
0441       <min-pool-size>6</min-pool-size>
0442       <max-pool-size>10</max-pool-size>
0443       <idle-timeout-minutes>15</idle-timeout-minutes>
0444 </tx-connection-factory>
0445
0446 <!-- ============================================= ======================
0447
0448
0449 <!-- ========================= T24 PW Designer (20020) ==================
0450 <!-- ============================================= ======================
0451
0452 </connection-factories>
0453
```

*(t24-ds.xml is a plain text file that of course can be seen using server OS tools or commands.)*

## 1.41 ASCII.VALUES, ASCII.VAL.TABLE

In the previous chapter the application field that accepted the underscore character was found after several attempts. Where is it set up?

Firstly we need to see T24 field data type in SS (i.e. STANDARD.SELECTION) application:

T24

```
Model Bank R14          STANDARD SELECTION FIELDS SEE

    FILE.NAME......... ABBREVIATION
--------------------------------------------------------------------------------
 11. 2 SYS.LANG.FIELD. N
 12. 2 SYS.GENERATED.. Y
  1. 3 SYS.FIELD.NAME. ORIGINAL.TEXT
  2. 3 SYS.TYPE....... D
  3. 3. 1 SYS.FIELD.NO 1
  4. 3. 1 SYS.VAL.PROG IN2A                <==== here
  ...
```

“IN2A” refers to a record in ASCII.VALUES application:

T24

```
Model Bank R14          ASCII Values SEE

    ID.............. IN2A
--------------------------------------------------------------------------------
  1 DEF.VALUE.TBL..... FRMB.STANDARD.A     STANDARD TABLE FOR IN2A FR
  2. 1 LANGUAGE....... GB                  English
  3. 1 LAN.VALUE.TBL.. STANDARD.A          STANDARD TABLE FOR IN2A
  2. 2 LANGUAGE....... FR                  French
  3. 2 LAN.VALUE.TBL.. FRMB.STANDARD.A     STANDARD TABLE FOR IN2A FR
 15 CURR.NO........... 6
 16. 1 INPUTTER....... 1_DL.RESTORE
 17. 1 DATE.TIME...... 28 MAY 14 14:46
 18 AUTHORISER........ 1_DL.RESTORE
 19 CO.CODE........... GB-001-0001        Model Bank R14
 20 DEPT.CODE......... 1                   Implementation




--------------------------------------------------------------------------------
08 OCT 2014 03:30:02  USER (22 APR) INPUTTER          [9637,INPAGE 1
ACTION
AWAITING PAGE INSTRUCTIONS
```

Default table for accepted characters can be seen in application ASCII.VAL.TABLE:

T24

```
Model Bank R14          ASCII Value Table SEE

    ID.............. FRMB.STANDARD.A
--------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. STANDARD TABLE FOR IN2A FR
  1. 2 FR DESCRIPTION. TABLE STANDARD POUR IN2A FR
  2. 1 START.RANGE.... 32
  3. 1 END.RANGE...... 94                  ^
  2. 2 START.RANGE.... 96
  3. 2 END.RANGE...... 122                 z
  2. 3 START.RANGE.... 188                 ij
  3. 3 END.RANGE...... 190                 ¿
  2. 4 START.RANGE.... 198                 Æ
  3. 4 END.RANGE...... 203                 Ë
  2. 5 START.RANGE.... 206                 Î
  3. 5 END.RANGE...... 207                 Ï
  2. 6 START.RANGE.... 219                 Û
  3. 6 END.RANGE...... 220                 Ü
  2. 7 START.RANGE.... 230                 æ
  3. 7 END.RANGE...... 235                 ë
  2. 8 START.RANGE.... 238                 î
  3. 8 END.RANGE...... 239                 ï
  2. 9 START.RANGE.... 251                 ?
  3. 9 END.RANGE...... 252                 ?
  4. 1 SINGLE.VALUE... 126                 ?
```

```
   4. 2 SINGLE.VALUE... 164                ?
   4. 3 SINGLE.VALUE... 176                ř
   4. 4 SINGLE.VALUE... 192                À
   4. 5 SINGLE.VALUE... 194                Â
   4. 6 SINGLE.VALUE... 212                Ô
   4. 7 SINGLE.VALUE... 217                Ù
   4. 8 SINGLE.VALUE... 224                à
   4. 9 SINGLE.VALUE... 226                â
   4.10 SINGLE.VALUE... 244                ô
   4.11 SINGLE.VALUE... 249                ?
   4.12 SINGLE.VALUE... 255                ?
   5. 1 GB ERR.MESSAGE. WRONG ALPHANUMERIC CHAR.
   5. 2 FR ERR.MESSAGE. CARACTERE ALPHANUMERIQUE INVALIDE
 ------------------------------------------------------------------------------
```

“WRONG ALPHANUMERIC CHAR” is what we saw in OFS output.  Underscore (ASCII 95) isn’t in the list.  Neither it presents in STANDARD.A record which is default for GB language.

For CONTEXT.ENQUIRY field ENQ.DESC T24 data type is “IN2ANY”:

T24

```
 Model Bank R14          STANDARD SELECTION FIELDS SEE

     FILE.NAME......... CONTEXT.ENQUIRY
 ------------------------------------------------------------------------------
...
   1.23 SYS.FIELD.NAME. ENQ.DESC
   2.23 SYS.TYPE....... D
   3.23. 1 SYS.FIELD.NO 15
   4.23. 1 SYS.VAL.PROG IN2ANY
...
```

...and the corresponding characters table accepts anything from ASCII 32 to ASCII 250.  Why not further?  Standard system delimiters are located there.  Besides @FM (also sometimes referred as @AM - attribute mark), @VM and @SM there’s also @TM (ASCII 251) - text mark (used to delimit output by FMT() function).

@TM usage example - cut the string in several chunks with the same length.  Unlike in FOLD() output, words might be cut in the middle:

jBC

```
the_phrase = \There's more than one way to say "non equal"...\
the_cut = FMT(the_phrase, '10L')
CHANGE @TM TO '|' IN the_cut
CRT the_cut
```

*(Backslash is used not only for line continuation but for string definition as well.)*

Output

```
There's mo|re than on|e way to s|ay "non eq|ual"...
```

## 1.42 IN2 routines

IN2 is not only the prefix for ASCII.VALUES records. There are core routines with same names that can be used to check user input or field contents in jBC routine.

See an example for date check (in case the T24 core hadn’t done that yet):

jBC

```
SUBROUTINE SUBR.0
* T24 mainline routine
$INSERT I_COMMON
$INSERT I_EQUATE

   COMI = '20141231'
   GOSUB DO.CHECK          ;* output: 31 DEC 2014

   COMI = '20150229'
   GOSUB DO.CHECK          ;* output: EB.RTN.DATE.DOES.NOT.EXIST

   COMI = '2014001'
   GOSUB DO.CHECK          ;* output: EB.RTN.INVALID.DATE.LENGTH

   COMI = '19171108'
   GOSUB DO.CHECK          ;* output: EB.RTN.YEAR.CANT.PRECEDE.1950

   COMI = '19171108'
   CALL IN2D(11, 'D' :@FM: 1000)
   GOSUB PRINT.RESULT            ;* output: 08 NOV 1917

   COMI = '171108'
   GOSUB DO.CHECK          ;* output: 17 NOV 2008

   COMI = '20141313'
   GOSUB DO.CHECK          ;* output: EB.RTN.MONTH.CAN..11

   COMI = '2014/12/31'
   GOSUB DO.CHECK          ;* output: 31 DEC 2014

   COMI = '30 NOV 2014'
   GOSUB DO.CHECK          ;* output: 30 NOV 2014

   COMI = '30 NOV 14'
   GOSUB DO.CHECK          ;* output: 30 NOV 2014

   RETURN
*-----------------------------------------------------------------------------
DO.CHECK:
   CALL IN2D(11, 'D')             ;* 11 is maximum allowed length of input
                                  ;* which was shown in '30 NOV 2014' example
   GOSUB PRINT.RESULT
   RETURN
```

```
PRINT.RESULT:
   IF ETEXT THEN TEXT = ETEXT
   ELSE TEXT = OCONV(ICONV(COMI, 'D'), 'D') ;* processed value in readable form
   CALL REM
   RETURN
*-----------------------------------------------------------------------------
END
```

To output the text instead of error code, we can use ERR subroutine in PRINT.RESULT section:

jBC - amended section PRINT.RESULT

```
PRINT.RESULT:
   IF ETEXT THEN
      E = ETEXT
      CALL ERR
      ETEXT = ''
      TEXT = 'ERROR OCCURRED'
   END ELSE
      TEXT = OCONV(ICONV(COMI, 'D'), 'D') ;* processed value in readable form
   END
   CALL REM
   RETURN
*-----------------------------------------------------------------------------
```

Error outputs:

T24

```
ACTION Y                                                DATE DOES NOT EXIST
CONTINUE (Y)                                                 ERROR OCCURRED
```

T24

```
ACTION Y                                              INVALID DATE LENGTH
CONTINUE (Y)                                                 ERROR OCCURRED
```

T24

```
ACTION Y                                          YEAR CANNOT PRECEDE 1950
CONTINUE (Y)                                                 ERROR OCCURRED
```

T24

```
ACTION Y                                            MONTH CAN BE (0)1-12
CONTINUE (Y)                                                 ERROR OCCURRED
```

Note:

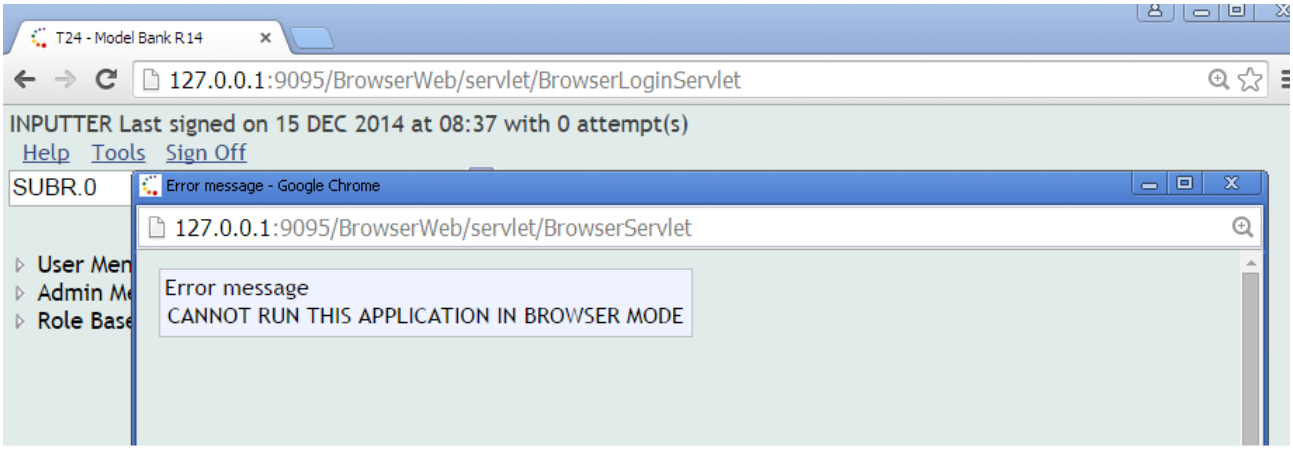


**Figure 27: Mainline T24 subroutine can't be run from Browser.**

## 1.43 Put the output from the previous chapter to the text file as a report

jBC has functionality to work with sequential files.

jBC

```
SUBROUTINE SUBR.0
* T24 mainline routine
$INSERT I_COMMON
$INSERT I_EQUATE

   GOSUB INIT
   IF the_err NE '' THEN
      TEXT = 'Error occurred: &' :FM: the_err
      CALL REM
      RETURN
   END

   COMI = '20141231'
   GOSUB DO.CHECK

   COMI = '20150229'
   GOSUB DO.CHECK

   COMI = '2014001'
   GOSUB DO.CHECK

   COMI = '19171108'
   GOSUB DO.CHECK

   COMI = '19171108'
   CALL IN2D(11, 'D' :@FM: 1000)
   GOSUB OUTPUT.RESULT

   COMI = '171108'
```

```
   GOSUB DO.CHECK

   COMI = '20141313'
   GOSUB DO.CHECK

   COMI = '2014/12/31'
   GOSUB DO.CHECK

   COMI = '30 NOV 2014'
   GOSUB DO.CHECK

   COMI = '30 NOV 14'
   GOSUB DO.CHECK

   CLOSESEQ f_out
   TEXT = 'Processing finished'   ;   CALL REM

   RETURN
*-----------------------------------------------------------------------------
INIT:
   the_err = ''
   dir_out = '&TTEMP&'   ;   file_out = 'report.txt'
   OPENSEQ dir_out, file_out TO f_out THEN    ;* file exists -
      WEOFSEQ f_out                           ;* truncate it
   END ELSE
      CREATE f_out ELSE the_err = 'file creation error' ; RETURN
   END

   RETURN

DO.CHECK:
   CALL IN2D(11, 'D')
   GOSUB OUTPUT.RESULT
   RETURN

OUTPUT.RESULT:
   the_err = ''
   the_output = COMI : ' ==> '

   IF ETEXT THEN the_output := ETEXT
   ELSE the_output := OCONV(ICONV(COMI, 'D'), 'D')
   WRITESEQ the_output TO f_outt ELSE
* In fatal cases we can use the following:
      TEXT = 'Write error to sequential file'
      CALL FATAL.ERROR(SYSTEM(40))   ;* don't need a RETURN here... we're off;
                                     ;* SYSTEM(40) returns subroutine name
   END

   RETURN
*-----------------------------------------------------------------------------
END
```

Let’s first check error handling, hence double “T” in “&TTEMP&”.  Compile...

jBC compilation

```
Warning: Variable f_outt is never assigned!
```

The error is intentional, don’t correct it yet.  Run:

T24

```
 ACTION
 CONTINUE (Y)                                Error occurred: file creation error
```

Correct the folder name (“&TEMP&”), try again:

jBASE fatal error

```
 -- LAST SIGN.ON, DATE: 15 DEC 2014     TIME: 10:50        ATTEMPTS: 0 --------
 15 DEC 2014 10:51:30  USER (22 APR) INPUTTER          [24886,IN]
 ACTION  ** Error [ NOT_FILE_VAR ] **
Variable is not an opened file descriptor , Line    55 , Source SUBR.0
Trap from an error message, error message name = NOT_FILE_VAR
Source changed to D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\ETC.BP\SUBR.0
0055 WRITESEQ the_output TO f_outt ELSE
jBASE debugger->
```

...our check never happened because the core failed even before that.  Anyway, FATAL.ERROR isn’t a good solution for online work.  We’re now in debugger but there’s almost nothing we can do, e.g.  to list some variables or all of them:

Debugger

```
V *
```

Output

```
...
...
  R.SPF.SYSTEM              : 20140328^Model Bank R14^O^OFF^../bnk.data^^^R14^1^^0^
                              200^^TAPE^Y^N^5^^../bnk.data^../bnk.run^../bnk.dict^^
                              NT^./backup.hd &M&^./restore.hd &M&^^5^500^^201404.00
                              2^^10^^^BEF]ESP]ITL]JPY]PTE]XAU]KWD]XAG^EURGBTMNS000^
                              20150630^^AA]AB]AC]AD]AI]AL]AM]AP]AR]BE]BL]BM]BR]CM]C
                              O]CR]DC]DD]DE]DL]DM]DS]DW]DX]EB]ET]EU]FA]FD]FE]FR]FT]
                              FW]FX]GP]IA]IB]IC]IF]IM]IN]IX]LC]LD]LI]LM]MC]MD]MF]MM
                              ]MO]MS]MT]NR]NS]OF]OO]OP]OV]PC]PD]PM]PO]PP]PR]PV]PW]R
                              C]RE]RP]RS]SA]SB]SC]SE]SF]SG]SL]SP]ST]SW]SY]T2]T3]T4]
                              T5]TK]TT]TX]VL]VP]VS]WR]WS]XT^^Y^N^50^^^^^^1^^^^^^^^^
                              ^^^^^^YES^^^^^^^^^^^^^^^^^^^^^^^^^^^55^25002_INPUTTE
                              R^1406111612^16185_AUTHORISER^GB0010001^1
  F.T24.SESSION             : File '..\bnk.data\eb\F_T24_SESSION'
  T.MULTI.TEXT              : 0
STANDARD Variables
  FM                        : (UNASSIGNED)
  the_output                : 20141231 ==> 31 DEC 2014
  f_out                     : File '&TEMP&\report.txt'
```

```
  PASSED.NO                 :
  f_outt                    : (UNASSIGNED)
  PASSED.CHAR               :
  the_err                   :
  dir_out                   : &TEMP&
  file_out                  : report.txt
```

Since we have T24 include files I_COMMON and I_EQUATE in our source, we can see global variables (like R.SPF.SYSTEM that we already know, global file variables like F.T24.SESSION etc). Then there are local variables like the_output that is ready to be written, f_out (that we failed to write to due to typo), the_err etc.

If this error happens under Browser, the user session will be timed out and the only trace will remain in JBoss log. If we also open another session and apply WHERE (V command to this one, we'll see something like:

Output

```
Port   Device     Account      PID      Command
8      VT100      r14          2512     Port 8 details at 11:02:22 15 DEC 2014
                                         Thread type Normal
                                         Usage Count 1M from 1M
                                         Application Open Files 63 , Actual O/S
                                         Open Files 57
                                         Memory: Free 0   Used 28,086,272
                                         @USER.ROOT INPUTTER-0
                                         READ's 66107 , WRITE's 288
                                         DELETE's 198 , CLEARFILE's 0
                                         EXECUTE's 122 , INPUT's 7322
                                         OPEN's 1924
                                    ---- jsh
                                         Program performing EXECUTE/PERFORM at
                                         jsh.b,111
                                    ---- EX
                                         Program in DEBUGGER at SUBR.0,55
```

To exit debugger type Q in its window. For Browser sessions debugger window can be opened using telnet connection to JBoss debugger port.

Now correct the typo, compile, run, see the output:

jsh

```
CT &TEMP& report.txt
```

Output

```
    report.txt
001 20141231 ==> 31 DEC 2014
002 20150229 ==> EB.RTN.DATE.DOES.NOT.EXIST
003 2014001 ==> EB.RTN.INVALID.DATE.LENGTH
004 19171108 ==> EB.RTN.YEAR.CANT.PRECEDE.1950
005 19171108 ==> 08 NOV 1917
006 20081117 ==> 17 NOV 2008
007 20141313 ==> EB.RTN.MONTH.CAN..11
008 20141231 ==> 31 DEC 2014
009 20141130 ==> 30 NOV 2014
010 20141130 ==> 30 NOV 2014
```

## 1.44 Put the output from the previous chapter to hashed file (jBASE style)

For T24 applications we use OPF to open, F.READ to read and F.WRITE to write. Here we have a temporary file that is created and updated directly.

Amended main section

```
// - commented   CLOSESEQ f_out
   CLOSE f_temp
   TEXT = 'Processing finished'   ;   CALL REM

   RETURN
```

Amended INIT section

```
*-----------------------------------------------------------------------------
INIT:
   OPEN 'F.TEMP' TO f_temp THEN
      the_err = ''
      CLEARFILE f_temp SETTING the_err
      IF the_err NE '' THEN RETURN
   END ELSE
      EXECUTE 'CREATE-FILE DATA F.TEMP 1 101 TYPE=J4'
      OPEN 'F.TEMP' TO f_temp ELSE the_err = 'File open error' ; RETURN
   END

   RETURN
```

Amended OUTPUT.RESULT section

```
OUTPUT.RESULT:
   IF ETEXT THEN the_output = ETEXT
   ELSE the_output = OCONV(ICONV(COMI, 'D'), 'D')

   rec_id = COMI
   WRITE the_output TO f_temp, rec_id

   RETURN
*-----------------------------------------------------------------------------
```

Results:

jsh

```
LIST F.TEMP

DICT F.TEMP...

20141231
20150229
2014001
19171108
20081117
20141313
20141130

 7 Records Listed
```

To see record contents without a dictionary:

jsh

```
LIST F.TEMP *A1
```

Output

```
DICT F.TEMP...    *A1...........

20141231          31 DEC 2014
20150229          EB.RTN.DATE.DO
                  ES.NOT.EXIST
2014001           EB.RTN.INVALID
                  .DATE.LENGTH
19171108          08 NOV 1917
20081117          17 NOV 2008
20141313          EB.RTN.MONTH.C
                  AN..11
20141130          30 NOV 2014
```

List results the same way as it was in the text file:

jsh

```
LIST F.TEMP EVAL "@ID:' ===> ':F1"
```

*(we can’t use A-correlative in EVAL)*

Output

```
DICT F.TEMP...    @ID:" ===> ":F1

20141231          20141231 ===>
                  31 DEC 2014
20150229          20150229 ===>
                  EB.RTN.DATE.DOE
                  S.NOT.EXIST
2014001           2014001 ===>
                  EB.RTN.INVALID.
                  DATE.LENGTH
19171108          19171108 ===>
                  08 NOV 1917
20081117          20081117 ===>
                  17 NOV 2008
20141313          20141313 ===>
                  EB.RTN.MONTH.CA
                  N..11
20141130          20141130 ===>
                  30 NOV 2014
```

F1 keyword helpfully comes from VOC:

jsh

```
CT VOC F1
```

Output

```
    F1
001 D
002 1
003
004
005 15T
006 S
```

*(so it outputs first data field)*

If we want the better format of the output without line wraps, use SELECT...[SAVING] EVAL:

jsh

```
SELECT F.TEMP EVAL "@ID:' ===> ':F1"
SAVE.LIST OUT
```

See the result (saved SELECT lists are actually plain text files in &SAVEDLISTS& directory):

jsh

```
CT &SAVEDLISTS& OUT
```

Output

```
OUT
001 20141231 ===> 31 DEC 2014
002 20150229 ===> EB.RTN.DATE.DOES.NOT.EXIST
003 2014001 ===> EB.RTN.INVALID.DATE.LENGTH
004 19171108 ===> 08 NOV 1917
005 20081117 ===> 17 NOV 2008
006 20141313 ===> EB.RTN.MONTH.CAN..11
007 20141130 ===> 30 NOV 2014
```

We also can copy records between hashed file and UD type.  There's &TEMP& directory in bnk.run that can be used for such manipulations:

jsh

```
COPY FROM F.TEMP TO &TEMP& ALL
LIST &TEMP&
```

Output

```
@ID

19171108
20081117
2014001
20141130
20141231
20141313
20150229
report.txt

 8 Records Listed
```

*(Besides copied records, there's report.txt that was created earlier.)*

We can edit any record in text editor or in JED...

jsh

```
JED &TEMP& 19171108
```

JED

```
File &TEMP& , Record '19171108'                          Insert      13:50:48
Command->
0001 08 NOV 1917
-------------------------------- End Of Record ---------------------------------
```

...amend it...

JED

```
File &TEMP& , Record '19171108'                              Insert      13:51:23
Command->
0001 08 NOV 1917 was actually 26 OCT
------------------------------- End Of Record --------------------------------

                ^^^^^^^^^^^^^^^^^^^^ appended text
```

...save, then copy back:

jsh

```
COPY FROM &TEMP& TO F.TEMP 19171108
```

Output

```
'19171108' already exists
0 records copied
```

To overwrite the record use OVERWRITING keyword:

jsh

```
COPY FROM &TEMP& TO F.TEMP 19171108 OVERWRITING
```

Output

```
1 records copied
```

See it:

jsh

```
LIST F.TEMP '19171108' *A1
```

Output

```
DICT F.TEMP...    *A1...........

19171108          08 NOV 1917 wa
                  s actually 26
                  OCT
```

Then we can delete unnecessary records:

jsh

```
SELECT &TEMP& UNLIKE "...'.txt'"
DELETE &TEMP&
```

Output

```
7 record(s) deleted.
```

Counting the number of records works here as well:

jsh

```
COUNT &TEMP&
```

Output

```
 1 Records counted
```

## 1.45 OFS log

Back to OFS. All OFS requests and responses can be found in F.OFS.REQUEST.DETAIL file - if OFS.SOURCE is set up accordingly.

“TAABS” record does not, so let’s copy it and amend:

T24

```
Model Bank R14          OFS SOURCE, SEE

    SOURCE.NAME....... TAABSLOG                     <=== new ID
------------------------------------------------------------------------------
  1 DESCRIPTION....... FOR TAG
  2 SOURCE.TYPE....... TELNET
  3. 1 LOGIN.ID....... ANY
 14 LOG.DETAIL.LEVEL.. NONE
 16 MAINT.MSG.DETS.... Y                            <=== new setting here...
 17 DET.PREFIX........ TLOG                         <=== and here
 24 SYNTAX.TYPE....... OFS
 26 GENERIC.USER...... INPUTTER
 42 CURR.NO........... 1
 43. 1 INPUTTER....... 22629_INPUTTER
 44. 1 DATE.TIME...... 15 DEC 14 14:06
 45 AUTHORISER........ 22629_INPUTTER
 46 CO.CODE........... GB-001-0001         Model Bank R14
 47 DEPT.CODE......... 1                   Implementation


------------------------------------------------------------------------------
15 DEC 2014 14:06:36  USER (22 APR) INPUTTER          [22629, PAGE 1
ACTION
AWAITING PAGE INSTRUCTIONS
```

Now, start tSS and repeat some earlier requests:

jsh

```
tSS TAABSLOG
```

OFS input

```
FT,TRAIN/I/VALIDATE,INPUTT/123456,FT140870BGYX,
```

OFS output

```
FT140870BGYX/TLOG143492116750974.01/-1/NO,PROCESSING.DATE:1:1=PROCESS DATE LESS
THAN TODAY
```

Try to commit a deal that finally passed the validation in earlier example:

OFS input

```
FT,TRAIN/I/PROCESS,INPUTT/123456,FT140870BGYX,PROCESSING.DATE::=20140422
```

OFS output

```
...
RECORD.STATUS:1:1=INAU,CURR.NO:1:1=1,INPUTTER:1:1=21167_INPUTTER__OFS_TAABSLOG_
FT140870BGYX,DATE.TIME:1:1=1412151411,CO.CODE:1:1=GB0010001,DEPT.CODE:1:1=1
```

See the log:

jsh

```
LIST F.OFS.REQUEST.DETAIL WITH @ID LIKE "'TLOG'..."
```

Output

```
...
TLOG143492116751074.00
TLOG143492116751076.04
TLOG143492116751068.01
TLOG143492116751080.04

 36 Records Listed
```

*(So many of them because this FT was done for AA arrangement and all deal-related messages were processed as a bulk.)*

Again “@” dictionary entry isn’t much populated so LIST isn’t an option...  See one of log records:

jsh

```
SELECT F.OFS.REQUEST.DETAIL WITH @ID LIKE "'TLOG'..." SAMPLE 1
CT F.OFS.REQUEST.DETAIL
```

Output

```
    TLOG1434921167510 83.00
001 AA.ARRANGEMENT.ACTIVITY
002
003 M
004 AAACT141127PHPFG5T
005
006 GB0010001
007 15 DEC 2014:071 14:11:23
008
009 15 DEC 2014:112 14:11:23
010 PROCESSED
011 AA.ARRANGEMENT.ACTIVITY,/M/PROCESS//1/POST,/******/GB0010001/////,AAACT141
    127PHPFG5T,UNAUTH
012 AAACT141127PHPFG5T/TLOG1434921167510 83.00/1,ARRANGEMENT:1:1=AA14083PY02L,A
    CTIVITY:1:1=LENDING-UPDATE.CASHFLOW-REPORTING,EFFECTIVE.DATE:1:1=20140422,
    CUSTOMER:1:1=100398,PRODUCT:1:1=MORTGAGE,CURRENCY:1:1=USD,LINKED.ACTIVITY:
    1:1=AAACT141127PHPFCWV,INITIATION.TYPE:1:1=SECONDARY,MASTER.AAA:1:1=AAACT1
    41127PHPFCWV,ACTIVITY.CLASS:1:1=LENDING-UPDATE.CASHFLOW-REPORTING,RECORD.S
    TATUS:1:1=INAU,CURR.NO:1:1=1,INPUTTER:1:1=21167_INPUTTER__OFS_TAABSLOG_FT1
    40870BGYX,DATE.TIME:1:1=1412151411,CO.CODE:1:1=GB0010001,DEPT.CODE:1:1=1
013 POST
014
015 1
016 21167
```

We can see OFS syntax in the field 11 (which stores the request); field 12 contains the response.  To find what was rejected use SELECT on a field STATUS:

jsh

```
SELECT F.OFS.REQUEST.DETAIL WITH STATUS NE "'PROCESSED'"
```

Oops...

Output

```
    !!! Error message Dictionary record STATUS is missing.
 not found !!!
```

We can either use “*A10” or find out field name; try the latter:

jsh

```
SELECT DICT F.OFS.REQUEST.DETAIL WITH *A2 EQ 10
```

Output

```
 No Records selected
```

Something is wrong...

jsh

```
COUNT DICT F.OFS.REQUEST.DETAIL
```

Output

```
 No Records counted
```

Surprise - dictionary is empty...  OK, let’s use A-correlative:

jsh

```
SELECT F.OFS.REQUEST.DETAIL WITH *A10 NE "'PROCESSED'"
CT F.OFS.REQUEST.DETAIL
```

Output

```
 2 Records selected

    TLOG143492116751073.01
001 AA.ARR.TERM.AMOUNT
...
010 ERROR
011 AA.ARR.TERM.AMOUNT,AA.PA/I/PROCESS//1/,/******/GB0010001/////,AA14083PY02L
    -COMMITMENT-20140422.1,ACTIVITY:1:1="LENDING-DISBURSE-COMMITMENT",ACTION:1
    :1="DRAW",
012 AA14083PY02L-COMMITMENT-20140422.1/TLOG143492116751073.01/-1/NO,AMOUNT:1:1
    =Disbursement of USD 3000.00 exceeds available amount of 0
...
...
    TLOG143492116750974.01
001 FT
002 TRAIN
003 I
004 FT140870BGYX
005 INPUTT
006 GB0010001
007 15 DEC 2014:563 14:09:34
008
009 15 DEC 2014:988 14:09:40
010 ERROR
011 FT,TRAIN/I/VALIDATE,INPUTT/******,FT140870BGYX,
012 FT140870BGYX/TLOG143492116750974.01/-1/NO,PROCESSING.DATE:1:1=PROCESS DATE
     LESS THAN TODAY
```

Due to the error in the first listed record, FT remained in “HOLD” status:

jsh

```
LIST FBNK.FUNDS.TRANSFER$NAU 'FT140870BGYX' RECORD.STATUS
```

Output

```
@ID.....................    RECORD.STATUS

FT140870BGYX                 IHLD
```

So far leave it as it is...

## 1.46 Overrides

Override in T24 contract input is the non-fatal error or warning that is to be confirmed by the user. In case of positive user feedback record is processed further and the overrdide is stored in application record, field OVERRIDE.

*(This field presents not in all applications, e.g. in ABBREVIATION it absents.)*

If we look at the application CURRENCY (which stores information about currencies and their rates), we can see it:

T24

```
Model Bank R14          CURRENCY SEE

    CURRENCY.CODE..... CHF                 Swiss Franc
------------------------------------------------------------------------------
 33 COUNTRY.CODE...... CH                  Switzerland
 41 CLS.CCY........... NO
 56. 1 OVERRIDE....... Current PERIODIC INTEREST record for CHF missing     <=====
 58 CURR.NO........... 3
 59. 1 INPUTTER....... 36247_INPUTTER__OFS_MB.LANG
 60. 1 DATE.TIME...... 02 JUN 14 15:51
 61 AUTHORISER........ 36247_INPUTTER_OFS_MB.LANG
 62 CO.CODE........... GB-001-0001         Model Bank R14
 63 DEPT.CODE......... 1                   Implementation









------------------------------------------------------------------------------
15 DEC 2014 16:15:00  USER (22 APR) INPUTTER          [29012,IPAGE 2
ACTION
AWAITING PAGE INSTRUCTIONS
```

Also, we can try to amend the rate.  In case of big difference between old and new rates T24 core will raise the override:

T24

```
Model Bank R14          CURRENCY INPUT

    CURRENCY.CODE..... CHF                 Swiss Franc
------------------------------------------------------------------------------
 16. 1 BUY.RATE....... 1.30
 17. 1 SELL.RATE...... 1.20
 18. 1 UPTO.SMALL.AMT.
 19. 1 TRSY.SMALL.SPRD
 20. 1 CUST.SMALL.SPRD
 21. 1 NEGOTIABLE.AMT. 88,428.10
 22. 1 TRSY.MED.SPREAD
 23. 1 CUST.MED.SPREAD
 24. 1 REVAL.RATE.....
 25. 1 TRSY.LIMIT.AMT.
 26 MIN.ROUND.AMOUNT..
 27 CASH.ONLY.ROUNDING
 28 MIN.ROUND.TYPE.... 1                   NATURAL ROUNDING
 29 CASH.ROUND.TYPE... 1                   NATURAL ROUNDING
 30. 1 DFT.CCY.UNIT...
 31 PRECIOUS.METAL.... NO
------------------------------------------------------------------------------
15 DEC 2014 16:17:15  USER (22 APR) INPUTTER          [29012,       VALIDATED
ACTION
OVERRIDE (Y/NO)                     41.36% DIFF TO AUTH MID RATE FOR CCY MKT 1 <====
```

Answer N, press F1, see that T24 warns you about losing your input (and that’s another override which isn’t stored though):

T24

```
------------------------------------------------------------------------------
15 DEC 2014 16:18:27  USER (22 APR) INPUTTER          [29012, PAGE 2   >>>5>>>
ACTION _
OVERRIDE (Y/NO)                                  YOU RETURN AND MAY LOSE INPUT
```

How it looks in Browser (override confirmation screen appears on “commit” and does not on “validate”):

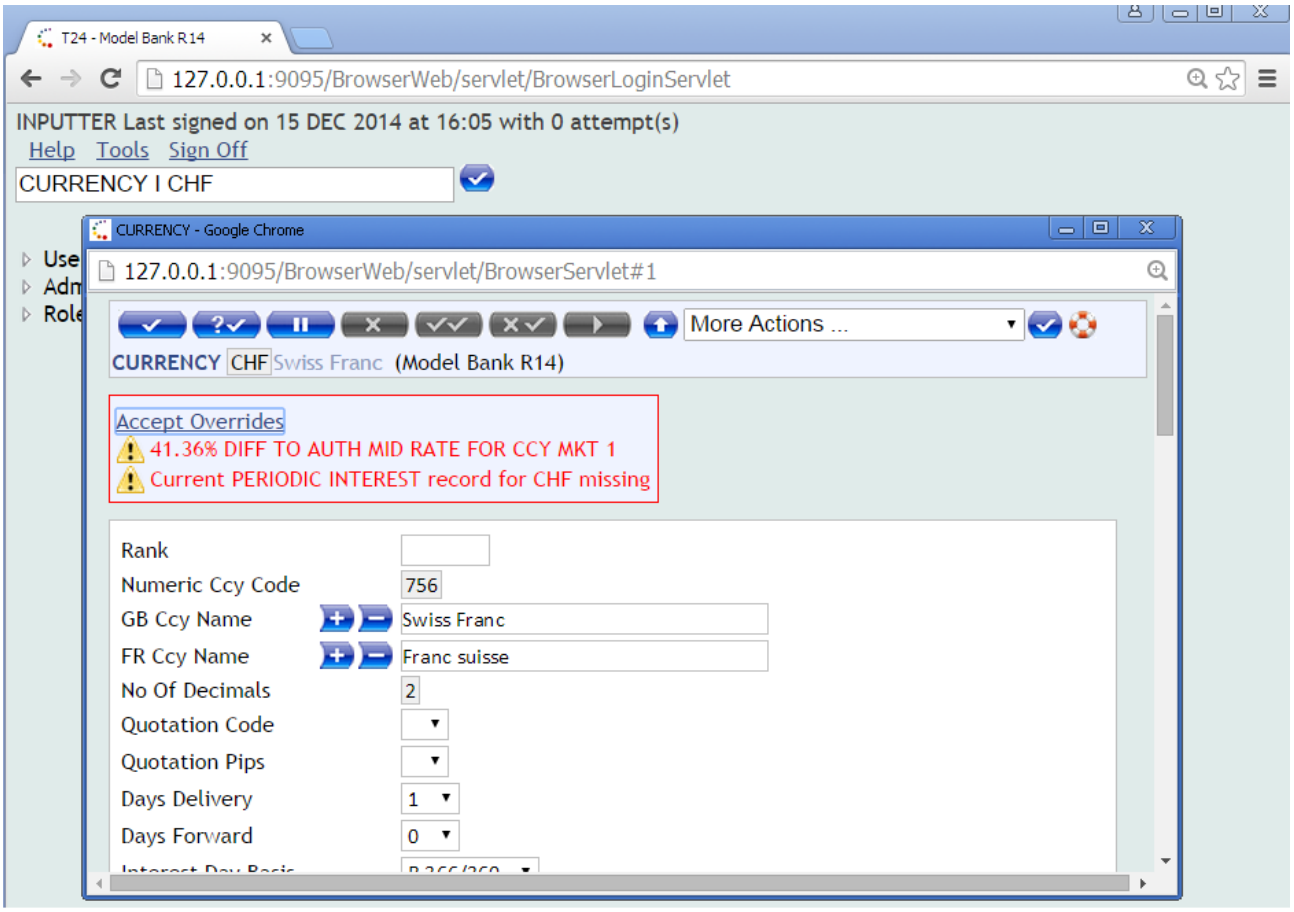


**Figure 28:  Overrides in Browser.**

Overrides are handled in OFS also.  There’s a field in VERSION - GTS.CONTROL - which handles overrides and error conditions.  It can be set up to accept overrides automatically, to put record on HOLD when override occurres, to reject record altogether when error occurres, to put record to HOLD when error occurres, to put record on HOLD even in case of success etc.

This setting also can be redefined in OFS message.  Example - value of 4 will put record on HOLD in any case:

OFS input

```
ABBREVIATION,/I/PROCESS/4/0,INPUTT/123456,SEESPF,ORIGINAL.TEXT::="COMPANY S BNK"
                           ^^^
```

OFS output

```
SEESPF//1,ORIGINAL.TEXT:1:1=COMPANY S BNK,RECORD.STATUS:1:1=IHLD,
                                          ^^^^^^^^^^^^^^^^^^^^^^^^
CURR.NO:1:1=3,INPUTTER:1:1=2583_INPUTTER__OFS_TAABS,DATE.TIME:1:1=1412151646,
CO.CODE:1:1=GB0010001,DEPT.CODE:1:1=1
```

There are settings for users concerning accepting of overrides.  T24 applications for that are OVERRIDE.CLASS and OVERRIDE.CLASS.DETAILS. For example, certain override has the different settings regarding the amount in question:

T24

```
Model Bank R14          OVERRIDE CLASS DETAILS SEE

    KEY............... ODRAFT.ALL
--------------------------------------------------------------------------------
  1. 1 DATA.DEF....... @MB.OCD.ODRAFT.ROUTINE()
  1. 2 DATA.DEF....... &2
  2. 1 CLASSIFICATION. MGR
  3. 1. 1 DATA.DEF.NO. 1
  4. 1. 1 COMPARISON.. EQ
  5. 1. 1 DATA.FROM... SAVINGS
  2. 2 CLASSIFICATION. AOFF
  3. 2. 1 DATA.DEF.NO. 2
  4. 2. 1 COMPARISON.. RG
  5. 2. 1 DATA.FROM... 10001
  6. 2. 1 DATA.TO..... 50000
  2. 3 CLASSIFICATION. BOFF
  3. 3. 1 DATA.DEF.NO. 2
  4. 3. 1 COMPARISON.. RG
  5. 3. 1 DATA.FROM... 50001
  6. 3. 1 DATA.TO..... 100000
--------------------------------------------------------------------------------
15 DEC 2014 16:37:54  USER (22 APR) INPUTTER         [21355,IPAGE 1   >>>2>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

If user doesn’t have sufficient rights to accept particular override, the record is saved but obtains the status “INAO”.

## 1.47  R.NEW, R.OLD and R.NEW.LAST

Earlier we addressed R.NEW - the global dimensioned array that contains the edited record data.  There are 2 more record “images” available.  If we create a VERSION for ABBREVIATION and attach a routine to it, we can use the created earlier record SEESPF to see the difference between these arrays.  The routine is an AUT.NEW.CONTENT one:

jBC

```
SUBROUTINE SUBR.ANC

$INSERT I_COMMON
$INSERT I_EQUATE

   R.NEW(AF) = 'Something new'

   DEBUG

   RETURN
END
```

Create the VERSION AB,TRAIN the same way FT,TRAIN was created:

T24

```
Model Bank R14          VERSION, INPUT

    PGM.NAME.VERSION.. ABBREVIATION,TRAIN
------------------------------------------------------------------------------
 49. 1 REKEY.FIELD.NO.
 50. 1 AUTOM.FIELD.NO. ORIGINAL.TEXT       ORIGINAL.TEXT
 51. 1 AUT.OLD.CONTENT
 52. 1 AUT.NEW.CONTENT @SUBR.ANC                APPL NOT DEFINED FOR THIS SUBR
 53. 1 MANDATORY.FIELD
 54 MULTI.POSSIBLE....
 55. 1. 1 VAL.ASSOC...
 56. 1. 1 SUB.ASSOC...
 57 LOCAL.REF.FIELD...
 58. 1 VALIDATION.FLD.
 59. 1 VALIDATION.RTN.
 60. 1 D.SLIP.FORMAT..
 61. 1 D.SLIP.FUNCTION
 62 D.SLIP.TRIGGER....
 63. 1 INPUT.ROUTINE..
 64. 1 AUTH.ROUTINE ..
------------------------------------------------------------------------------
15 DEC 2014 17:02:23  USER (22 APR) INPUTTER          [29393,IPAGE 4   >>>7>>>
ACTION
```

To be able to commit this record firstly put it to hold (HLD) not to lose the input, then amend the PGM.FILE record:

T24

```
Model Bank R14          PROGRAM FILE, INPUT

    PROGRAM            SUBR.ANC
------------------------------------------------------------------------------
  1 TYPE.............. S
  2. 1 GB SCREEN.TITLE
  3 ADDITIONAL.INFO...
  4. 1 BATCH.JOB......
  5 PRODUCT........... EB
  6 SUB.PRODUCT.......
  7. 1 DESCRIPTION....
  8. 1 APPL.FOR.SUBR.. ABBREVIATION        Abbreviation    <==== add this value here
  8. 2 APPL.FOR.SUBR.. FUNDS.TRANSFER      FUNDS.TRANSFER
  9 ACTIVATION.FILE...
 10 MT.KEY.COMPONENT..
 11 MT.KEY.FILE.......
 12 REC.VERIFY........
 13 BYPASS.SEL........
 14 BULK.NO...........
 15 JOB.RATING........
------------------------------------------------------------------------------
15 DEC 2014 17:04:25  USER (22 APR) INPUTTER          [29393,IPAGE 1   >>>3>>>
ACTION
```

Open record SEESPF with it (AB,TRAIN I SEESPF). As soon as we confirm the override “RECORD IN HOLD-STATUS” we fall into debugger:

T24

```
 ------------------------------------------------------------------------------
 15 DEC 2014 17:06:49  USER (22 APR) INPUTTER          [13145,IN]
 ACTION
 DEBUG statement seen
0004     DEBUG
jBASE debugger->
```

See the difference between 3 record images using field 1:

Debugger

```
jBASE debugger->V R.NEW(1)
COMMON variables
  R.NEW(1)                  : Something new
jBASE debugger->V R.OLD(1)
COMMON variables
  R.OLD(1)                  : SPF S SYSTEM
jBASE debugger->V R.NEW.LAST(1)
COMMON variables
  R.NEW.LAST(1)             : COMPANY S BNK
jBASE debugger->
```

So, R.NEW is the edited record (with changes), R.OLD is last authorised record. R.NEW.LAST is last unauthorised record as it was before editing.

Other global variables of interest:

Debugger

```
jBASE debugger->V ID.NEW
COMMON variables
  ID.NEW                    : SEESPF
jBASE debugger->V OPERATOR
COMMON variables
  OPERATOR                  : INPUTTER
jBASE debugger->V TNO
COMMON variables
  TNO                       : 15779
jBASE debugger->V TODAY
COMMON variables
  TODAY                     : 20140422
```

Application fields properties:

Debugger

```
jBASE debugger->V F(1)
COMMON variables
  F(1)                      : ORIGINAL.TEXT.....
jBASE debugger->V N(1)
COMMON variables
  N(1)                      : 54.3
jBASE debugger->V T(1)
COMMON variables
  T(1)                      : A
```

54.3 is maximum/minimum length of input:

T24 ABBREVIATION input

```
   1 ORIGINAL.TEXT..... 12                                    TOO FEW CHARACTERS
```

“A” in T array means “IN2A” check.

*(Some - though a bit old - explanations for T24 global variables and arrays can be found in T24.BP, file I_RULES.)*

To see call stack in debugger, use j -g command:

Debugger

```
jBASE debugger->j -g
Backtrace:
#0: jmainfunction.b:5
#1: EX:50 -> Line 50 , Source EX
#2: T.EX:265 -> Line 265 , Source T.EX
#3: RUN.APPLICATION:77 [0] -> Line 77 , Source RUN.APPLICATION
#3: RUN.APPLICATION:145 -> Line 145 , Source RUN.APPLICATION
#4: EB.EXECUTE.APPLICATION:55 -> Line 55 , Source EB.EXECUTE.APPLICATION
#5: ABBREVIATION:58 -> Line 58 , Source ABBREVIATION
#6: RECORD.READ:1478 -> Line 1478 , Source RECORD.READ
#7: SUBR.ANC:5 -> Line 5 , Source RECORD.READ

Backtrace log:
Line 0 , Source jmainfunction.b , Level 0
Line 50 , Source EX
Line 265 , Source T.EX
Line 77 , Source RUN.APPLICATION
Line 145 , Source RUN.APPLICATION
Line 55 , Source EB.EXECUTE.APPLICATION
Line 58 , Source ABBREVIATION
Line 1478 , Source RECORD.READ
```

## 1.48 History

Many T24 applications have history where either changed or reversed records go. See how many's there:

jsh

```
COUNT F.PGM.FILE WITH TYPE EQ 'H'
```

Output

```
 1299 Records counted
```

Let's take ACCOUNT and see history records in more detail.

jsh

```
LIST ONLY FBNK.ACCOUNT$HIS
```

Output

```
69612;3
71412;1
USD140100001;1
USD141850001;2
USDGBP190210001;1
USD140550001;3
69663;3
69647;1
...
```

*(Records that have numeric @ID are customer accounts, ones that start with currency @ID - internal accounts like suspense accounts, cash tills etc.)*

Which records were put to history, say, at least 3 times?

jsh

```
LIST ONLY FBNK.ACCOUNT$HIS WITH EVAL "FIELD(@ID,';',2)" EQ 3
```

Output

```
69612;3
USD140550001;3
69663;3
69582;3
71293;3
69598;3
69493;3
68721;3
69485;3
69507;3
69558;3
```

```
71196;3
71331;3
69547;3
69523;3
69469;3

 16 Records Listed
```

The same output can also be achieved using the following query...

jsh

```
LIST ONLY FBNK.ACCOUNT$HIS WITH EVAL "@ID[-2,2]" EQ ';3'
```

...or the following one:

jsh

```
LIST ONLY FBNK.ACCOUNT$HIS WITH @ID LIKE "0X';3'"
```

But was it a change or something else?

jsh

```
LIST FBNK.ACCOUNT$HIS RECORD.STATUS WITH EVAL "FIELD(@ID,';',2)" EQ 3
                      ^^^^^^^^^^^^^
```

Output

```
@ID.............    RECORD.STATUS

            69612;3
    USD140550001;3    CLOSED
            69663;3
            69582;3
            71293;3    CLOSED
            69598;3
            69493;3
            68721;3    CLOSED
            69485;3
            69507;3
            69558;3
            71196;3    CLOSED
            71331;3    CLOSED
            69547;3
            69523;3
            69469;3

 16 Records Listed
```

Accounts that are closed are no more in LIVE file:

jsh

```
SELECT FBNK.ACCOUNT$HIS WITH EVAL "FIELD(@ID,';',2)" EQ 3 SAVING EVAL
   "FIELD(@ID,';',1)"
LIST ONLY FBNK.ACCOUNT
```

Output

```
69612
 ** Error [ 202 ] **
Record 'USD140550001' is not on file.
69663
69582
 ** Error [ 202 ] **
Record '71293' is not on file.
69598
69493
 ** Error [ 202 ] **
Record '68721' is not on file.
69485
69507
69558
 ** Error [ 202 ] **
Record '71196' is not on file.
 ** Error [ 202 ] **
Record '71331' is not on file.
69547
69523
69469

 11 Records Listed
```

To suppress error messages use “(R” option:

jsh

```
SELECT FBNK.ACCOUNT$HIS WITH EVAL "FIELD(@ID,';',2)" EQ 3 SAVING
   EVAL "FIELD(@ID,';',1)"
LIST FBNK.ACCOUNT CURR.NO (R
```

Output

```
@ID.............    CURR.NO

              69612          6
              69663          4
              69582          6
              69598          5
              69493         10
              69485          7
              69507          6
              69558          5
              69547          6
              69523          5
              69469          5
```

List all variants of record @ID tails:

jsh

```
SELECT FBNK.ACCOUNT$HIS WITH EVAL "FIELD(@ID,';',2)" AS tail SAVING UNIQUE tail

 9 Records selected

>SAVE.LIST TAILS
9 record(s) saved to list 'TAILS'
jsh r14  -->EDIT.LIST TAILS
.jBASE.el.8
TOP
.P
TOP
001 3
002 1
003 2
004 8
005 5
006 6
007 4
008 9
009 7
BOTTOM
.EXIT
Record '.jBASE.el.8' exited from file '.'
List 'TAILS' exited
```

*(EDIT.LIST invokes old Universe-style editor; JED is more convenient one so we could use JED &SAVEDLISTS& TAILS instead.)*

*(Universe is another multi-value DBMS that Temenos used before jBASE.)*

If we need it, we can sort this list:

jsh

```
SORT.LIST TAILS
CT &SAVEDLISTS& TAILS
```

Output

```
List 'TAILS' sorted with 9 records

    TAILS
001 1
002 2
003 3
004 4
005 5
006 6
007 7
008 8
009 9
```

Which records were amended mostly?  Those that have 9 in the tail:

jsh

```
LIST FBNK.ACCOUNT$HIS RECORD.STATUS WITH EVAL "FIELD(@ID,';',2)" EQ 9
```

Output

```
69493;9
```

We can try to amend this record in T24 to put “;10” to history:

T24 “AWAITING APPLICATION” prompt

```
AC, I 69493
```

T24

```
 ------------------------------------------------------------------------------
 15 DEC 2014 21:34:55  USER (22 APR) INPUTTER          [23491,IN]
 ACTION 69493                                       No input for an AA account
 AWAITING ID
```

Well...  try to find non-AA one:

AWAITING ID

```
L L
```

T24

```
 Model Bank R14          ACCOUNT, LIST
                                    USD  Savings Account

 ------------------------------------------------------------------------------
        SELECTION FIELDS - LIVE FILE      [EQ NE LT GT LE GE RG NR LK UL]
 ------------------------------------------------------------------------------
 0.. @ID                    0A. ACCOUNT.NUMBER        0B. SHOW.TILL.ACS
 0C. START.DATE             0D. END.DATE              0E. CAP.DATE
 0F. LONG.POS.SIGN          0G. VAL.DATE              0H. TXN.AMT
 0I. TXN.CODE               0J. MERGE.NCU             0K. AC.NO.DAYS
 1.. CUSTOMER               2.. CATEGORY              2A. PRODCAT
 3.. ACCOUNT.TITLE.1        4.. ACCOUNT.TITLE.2       5.. SHORT.TITLE
 6.. MNEMONIC               7.. POSITION.TYPE         8.. CURRENCY
 9.. CURRENCY.MARKET        10. LIMIT.REF             11. ACCOUNT.OFFICER
 12. OTHER.OFFICER          13. POSTING.RESTRICT      14. RECONCILE.ACCT
 15. INTEREST.LIQU.ACCT     16. INTEREST.COMP.ACCT    17. INT.NO.BOOKING
 18. REFERAL.CODE           19. WAIVE.LEDGER.FEE      20. LOCAL.REF
 21. CONDITION.GROUP        22. INACTIV.MARKER        23. OPEN.ACTUAL.BAL
 24. OPEN.CLEARED.BAL       25. ONLINE.ACTUAL.BAL     26. ONLINE.CLEARED.BAL
 27. WORKING.BALANCE        28. DATE.LAST.CR.CUST     29. AMNT.LAST.CR.CUST
 ------------------------------------------------------------------------------
 15 DEC 2014 21:36:13  USER (22 APR) INPUTTER          [23491,INPAGE  1 >>7>>>
 ACTION _
 ENTER SELECTION (eg CUST EQ 123456) , <F5> TO EXECUTE LIST
```

Press F3/F2 to navigate through pages in fields list.  At the bottom page we can see not only data fields (starting from numbers, e.g.  “201 OVERRIDE”) but also so-called “I-descriptors” (e.g.  “XI. RESIDENCE”). Later about them; our field of interest is at the page 6 and it’s “181 ARRANGEMENT.ID”. Type at the “ACTION” line: 181 EQ ” Enter.  The result:

T24

```
Model Bank R14          ACCOUNT, LIST
                                    USD  Savings Account

------------------------------------------------------------------------------
       SELECTION FIELDS - LIVE FILE      [EQ NE LT GT LE GE RG NR LK UL]
------------------------------------------------------------------------------
156 FORWARD.MVMTS         157 CREDIT.CHECK          158 AVAILABLE.BAL.UPD
159 CONSOLIDATE.ENT       160 MAX.SUB.ACCOUNT       161 MASTER.ACCOUNT
162 LOCK.INC.THIS.MVMT    163 CLOSED.ONLINE         164 NEXT.AF.DATE
165 NEXT.ACCT.CAP         166 NEXT.EXP.DATE         167 DATE.LAST.UPDATE
168 NEXT.STMT.DATE        169 EXPOSURE.DATES        170 PORTFOLIO.NO
171 SHADOW.ACCOUNT        172 FWD.ENTRY.HOLD        173 FIRST.AF.DATE
174 CASH.POOL.GROUP       175 OPEN.ASSET.TYPE       176 LAST.COM.CHG.DATE
177 IC.CHARGE.ID          178 IC.NEXT.CAP.DATE      179 IC.PRODUCT
180 IC.LST.PROD.CAP       181 ARRANGEMENT.ID        182 ACC.DEB.LIMIT
183 MANDATE.APPL          184 MANDATE.REG           185 MANDATE.RECORD
186 DR.ADJ.AMOUNT         187 DR2.ADJ.AMOUNT        188 CR.ADJ.AMOUNT
189 CR2.ADJ.AMOUNT        190 EVENT                 191 FIELD
------------------------------------------------------------------------------
ARRANGEMENT.ID     EQ ''              <=== field number was replaced to field name
------------------------------------------------------------------------------
15 DEC 2014 21:36:13  USER (22 APR) INPUTTER          [23491,INPAGE  6 >>8>>>
ACTION
ENTER SELECTION (eg CUST EQ 123456) , <F5> TO EXECUTE LIST
```

We can also add another selection:  CURR.NO GT 7.  Then Press F5:

T24

```
Model Bank R14


------------------------------------------------------------------------------

         No records were found that matched the selection criteria

         SSELECT FBNK.ACCOUNT WITH   ARRANGEMENT.ID = ""
         AND   CURR.NO GT "7" AND   CO.CODE = "GB0010001"
         BY CATEGORY BY CURRENCY









------------------------------------------------------------------------------
```

Here we actually executed so-called “enquiry” with some default settings.  Not only we are limited to accounts belonging to current branch but here’s also SSELECT (meaning sort by @ID) together with “BY CATEGORY BY CURRENCY” sort that really can’t be very fast on hugely populated tables - SSELECT firstly sorts records by @ID and then they are sorted another time:

jsh

```
SSELECT FBNK.ACCOUNT BY CURRENCY BY CUSTOMER

 1385 Records selected

>LIST FBNK.ACCOUNT CURRENCY CUSTOMER
```

Output

```
@ID.................    CURRENCY    CUSTOMER..

              22233    AUD             100773
   CAD1091500010001    CAD
   CAD1402700010001    CAD
   CAD1591500010001    CAD
CADUSD1401600010001    CAD
              69078    CAD             100272
              69019    CAD             100342
              22241    CAD             100774
              66327    CAD             128081
              66575    CAD             188919
   CHF1280000020001    CHF
...
```

*(Absolutely the same result when we use SELECT instead of SSELECT.)*

To continue selection of ACCOUNT we’re switching to another branch (COMPANY in T24 terms) using T24 command PASSWORD:

T24

```
 Model Bank R14          PASSWORD


 ------------------------------------------------------------------------------
 SELECT ACTIVATION
 1 = CHANGE PASSWORD
 2 = CHANGE COMPANY
 3 = DEACTIVATE PROFILE

...




 -- LAST SIGN.ON, DATE: 15 DEC 2014     TIME: 21:34        ATTEMPTS: 0 --------
 15 DEC 2014 23:33:06  USER (22 APR) INPUTTER          [23978,IN]
 ACTION _
 AWAITING ACTIVATION CODE (1-3)
```

Choose option 2:

T24

```
ACTION _
PLEASE KEY IN COMPANY CODE
```

Type COMPANY @ID or mnenonic, e.g.  EU1; the screen header shows the change:

T24

```
 Model Bank - Europe     SELECT APPLICATION


 ------------------------------------------------------------------------------
...
```

Now repeat the account search:

T24

```
AC L L
```

Selection criteria

```
181 EQ ''
CURR.NO GT 7
```

Result:

T24

```
 Model Bank - Europe


 ------------------------------------------------------------------------------

          No records were found that matched the selection criteria

          SSELECT FEU1.ACCOUNT WITH   ARRANGEMENT.ID = ""
          AND   CURR.NO GT "7" BY CATEGORY BY CURRENCY









 ------------------------------------------------------------------------------
 15 DEC 2014 23:40:41  USER (22 APR) INPUTTER          [23978,IPAGE   1 >>>1>>>
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

We see that account file is different (FEU1.ACCOUNT). Since application ACCOUNT has FIN type as per FILE.CONTROL, file prefix (after initial “F”) is company mnemonic.

However, when we try CU L L for the same company (with some improbable selection like @ID UL ...  that means “unlike anything”), it shows FBNK.CUSTOMER as customer file:

T24

```
     No records were found that matched the selection criteria

     SSELECT FBNK.CUSTOMER WITH   @ID UNLIKE "..." BY
     MNEMONIC
```

Why not FEU1.CUSTOMER? The answer is that when a new company is created, it’s decided, among other options, whether customers will be at the same file or not.  This setting is then written to COMPANY record:

T24

```
 Model Bank - Europe     COMPANY SEE

     COMPANY.CODE...... EU-001-0001         EU Countries   COMPANY GROUP 1
 ------------------------------------------------------------------------------
   1. 1 GB COMPANY.NAME Model Bank - Europe
   2. 1 NAME.ADDRESS... Model Bank - Europe
   3 MNEMONIC.......... EU1
...
  18 CUSTOMER.COMPANY.. GB-001-0001
  19 CUSTOMER.MNEMONIC. BNK
```

Return to main branch, leave only first selection criterion (ARRANGEMENT.ID EQ ”), see the output:

T24

```
 Model Bank R14
 1          66575  THOMRECAD  Corporate| Current Account CAD
 2          11002  COCACOLCHF Private C| Current Account CHF
 3          11509  BRANSONCHF Private C| Current Account CHF                  -1,68
 4          13323  GENMOTOCHF Treasury | Current Account CHF                 -14,00
 5          14028  CITIGROCHF Credit Ma| Current Account CHF
 6          14378  MICROSOCHF Private C| Current Account CHF                  23,24
 7          66419  CORPACC3   Trade Fin| Current Account CHF
 8          66559  THOMRECHF  Corporate| Current Account CHF
 9          67083  HEWWILLCHF Private C| Current Account CHF
10          67091  GWARDCHF   Private C| Current Account CHF
11          68551  WILLNORCHF Treasury | Current Account CHF
12          10968  BUFFETEUR  Private C| Current Account EUR                   1,66
13          11018  COCACOLEUR Private C| Current Account EUR                  48,52
14          11215  NIKEEUR    Corporate| Current Account EUR
15          11312  JOHNSONEUR Trade Fin| Current Account EUR                 -40,32
16          11517  BRANSONEUR Private C| Current Account EUR                  13,40
17-         11727--SAINSEUR  -Treasury |-Current Account-EUR--                35,34
185         130373:PEUGEOTEUR2Treasury |UCurrent Account2EUR1, PAGE   1 >>>3>>>
19C         13331  GENMOTOEUR Treas
9AWAITING PAGE INSTRUCTIONS
```

The screen is sort of corrupted at the bottom because nowadays nobody bothers to think about terminal mode...  Edit the record 11002; amend the field 3:

T24

```
Model Bank R14          ACCOUNT, INPUT
                                    CHF  Current Account
    ACCOUNT.NUMBER.... 11002             COCA-COLA
--------------------------------------------------------------------------------
  1 CUSTOMER.......... 100282            Coca-cola
  2 CATEGORY.......... 1-001             Current Account
  3 ACCOUNT.TITLE.1... COLA-LOCA                                        <=====
  4 ACCOUNT.TITLE.2...
  5 SHORT.TITLE....... COCA-COLA
  6 MNEMONIC.......... COCACOLCHF
  7 POSITION.TYPE..... TR                TRADING POSITION
  8 CURRENCY.......... CHF               Swiss Franc
  9 CURRENCY.MARKET... 1                 Currency Market
 10 LIMIT.REF.........
 11 ACCOUNT.OFFICER... 14                Private Corporate Action Officer
 12. 1 OTHER.OFFICER..
 13. 1 POSTING.RESTRIC
 14 RECONCILE.ACCT....
 15 INTEREST.LIQU.ACCT
 16 INTEREST.COMP.ACCT
--------------------------------------------------------------------------------
15 DEC 2014 21:56:24  USER (22 APR) INPUTTER          [23491,I      VALIDATED
ACTION
```

Commit the record, open it in SEE mode.  Then press F1 and input ; at “AWAITING ID” prompt:

T24

```
Model Bank R14          ACCOUNT, SEE
                                    CHF  Current Account
    ACCOUNT.NUMBER.... 11002 ;          COLA-LOCA
--------------------------------------------------------------------------------
  1 CUSTOMER.......... 100282            Coca-cola
  2 CATEGORY.......... 1-001             Current Account
  3 ACCOUNT.TITLE.1... COLA-LOCA                              <==== new content
                       COCA-COLA                              <==== old content
  5 SHORT.TITLE....... COCA-COLA
  6 MNEMONIC.......... COCACOLCHF
  7 POSITION.TYPE..... TR                TRADING POSITION
  8 CURRENCY.......... CHF               Swiss Franc
  9 CURRENCY.MARKET... 1                 Currency Market
 11 ACCOUNT.OFFICER... 14                Private Corporate Action Officer
 21 CONDITION.GROUP... 2                 Current Account Others
 46. 1 CAP.DATE.CHARGE 31 MAR 2014
 47. 1 CAP.DATE.CR.INT 31 MAR 2014
 48. 1 CAP.DATE.C2.INT 31 MAR 2014
 49. 1 CAP.DATE.DR.INT 31 MAR 2014
 50. 1 CAP.DATE.D2.INT 31 MAR 2014
--------------------------------------------------------------------------------
15 DEC 2014 22:00:15  USER (22 APR) INPUTTER          [23491, PAGE 1   >>>3>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

Here we can see changes.  If we want to see earlier changes, we can supply record @ID as “;n”, where n is history record number (only one change - ours - was done for this account so we can try to see the mostly changed one that we discovered before):

T24

```
Model Bank R14          ACCOUNT, SEE
                                    USD  Savings Account
    ACCOUNT.NUMBER.... 69493 ;           Savings Account
------------------------------------------------------------------------------
 95 INTEREST.CCY...... USD               US Dollar
 96 INTEREST.MKT...... 1                 Currency Market
 99. 1 ALT.ACCT.TYPE.. LEGACY
 99. 2 ALT.ACCT.TYPE.. T24.IBAN
100. 2 ALT.ACCT.ID.... GB40NWBK60161300069493
 99. 3 ALT.ACCT.TYPE.. PREV.IBAN
108 ALLOW.NETTING..... NO
141 HVT.FLAG.......... NO                                          <=== here

157 CREDIT.CHECK...... AVAILABLE
...
```

T24

```
Model Bank R14          ACCOUNT, SEE
                                    USD  Savings Account
    ACCOUNT.NUMBER.... 69493 ; 9         Savings Account
------------------------------------------------------------------------------
 95 INTEREST.CCY...... USD               US Dollar
 96 INTEREST.MKT...... 1                 Currency Market
 99. 1 ALT.ACCT.TYPE.. LEGACY
 99. 2 ALT.ACCT.TYPE.. T24.IBAN
100. 2 ALT.ACCT.ID.... GB40NWBK60161300069493
 99. 3 ALT.ACCT.TYPE.. PREV.IBAN
108 ALLOW.NETTING..... NO
141 HVT.FLAG.......... NO                Defaulted by System       <=== and here

157 CREDIT.CHECK...... AVAILABLE
158 AVAILABLE.BAL.UPD. DEBITS
181 ARRANGEMENT.ID.... AA140839YNHB
190. 1 EVENT.......... DD.FAILED         DD Failed                 <=== and here
                       DD.EXECUTED       DD Executed
200. 1 REQUEST.ID..... EBAR1408348872                              <=== and here
                       EBAR1408337102
------------------------------------------------------------------------------
15 DEC 2014 22:07:08  USER (22 APR) INPUTTER          [23491, PAGE 2   >>>6>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

And so on and so forth...

T24

```
Model Bank R14          ACCOUNT, SEE
                                    USD  Savings Account
    ACCOUNT.NUMBER.... 69493 ; 8         Savings Account
------------------------------------------------------------------------------
 95 INTEREST.CCY...... USD               US Dollar
 96 INTEREST.MKT...... 1                 Currency Market
 99. 1 ALT.ACCT.TYPE.. LEGACY
 99. 2 ALT.ACCT.TYPE.. T24.IBAN
100. 2 ALT.ACCT.ID.... GB40NWBK60161300069493
 99. 3 ALT.ACCT.TYPE.. PREV.IBAN
108 ALLOW.NETTING..... NO
141 HVT.FLAG.......... NO                Defaulted by System

157 CREDIT.CHECK...... AVAILABLE
158 AVAILABLE.BAL.UPD. DEBITS
181 ARRANGEMENT.ID.... AA140839YNHB
190. 1 EVENT.......... DD.EXECUTED       DD Executed
                       DD.CANCEL         DD Mandate cancelled
200. 1 REQUEST.ID..... EBAR1408337102
                       EBAR1408389805
------------------------------------------------------------------------------
15 DEC 2014 22:08:16  USER (22 APR) INPUTTER          [23491, PAGE 2   >>>5>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

## 1.49 ENQUIRY

T24 application ENQUIRY contains enquiries that can be used to list T24 data according to user selection criteria.  When we use commands like AC L or L L - the default (so-called “percent enquiry”) is launched (in this example - “%ACCOUNT”).

In the previous chapter we saw the corrupted screen.  List of unauthorised records looks differently but still corrupted...

See the output of AC E command in T24:

T24

```
 Model Bank R14          Account - Exception List

   Account No     Short Title                          Mnemonic   StatuInputter
 ------------------------------------------------------------------------------
 1         21067  ALLEGRA                                         INAU 82421_OFFICER__
 2         66028                                                  IHLD 93663_BASEUSER_
 3         66036                                                  IHLD 93663_BASEUSER_
 4         67938  Lindsay Fox                          FOXLUSD1   INAU 62095_OFFICER__
 5         67989  Kerry Packer                         PACKMDUSD2 INAU 62095_OFFICER__
 6         68071  Dell Computer                        DELLUSDMD  INAU 62095_OFFICER__
 7         69442  Current Account                      GERLIUSDCA INAU 355_CSAGENT1__0
 8         71846  SEARS USD AC                         SEARSRCA   INAU 37540_OFFICER__
 9         72036  Constlln Energy Ac                              INAU 77911_OFFICER__
10         72068  Constlln Energy Ac                              INAU 77911_OFFICER__
11         72524  Stephan Cur Account                  STEPCUAC   INAU 62262_OFFICER__
12         66017                                                  IHLD 93663_BASEUSER_
13         72532  Savings Account                      SAVROB     INAU 62262_OFFICER__
14         72548  Francois Savings Account             FRANSAV    INAU 62262_OFFICER__
15         72559  Reg Savings Account                  SAVACREG   INAU 62262_OFFICER__
16         71145  9 Month Term Deposit                 GERLIDEP   INAU 355_CSAGENT1__0
17-        71347--3 Year Term Deposit                 -HEINEDEP  -INAU-355_BUILDUSER81
185          US|31INTERNAL TILL ACCOUNT                 TILL58L  PAGE   1 >>>1>>>
19C          EU|  EURTCPURVA                            EURTC
2AWAITING PAGE INSTRUCTIONS
```

This is because corresponding enquiry is also different; check its paging setting:

jsh

```
LIST F.ENQUIRY '%ACCOUNT$NAU' PAGE.SIZE
```

Output

```
@ID.............................................    PAGE.SIZE

%ACCOUNT$NAU                                       4,99
```

PAGE.SIZE is the reason why corruption happens:

T24

```
 Model Bank R14          ENQUIRY SEE

     ENQUIRY........... %ACCOUNT
 ------------------------------------------------------------------------------
   1 PAGE.SIZE ........ 4,99
   2 FILE.NAME......... ACCOUNT
   4. 1 FIXED.SORT..... CATEGORY
   4. 2 FIXED.SORT..... CURRENCY
...
```

The normal setting for PAGE.SIZE is 4,19 (meaning that data is to be output in terminal lines 4 to 19, counting from 0).  Find where it's set up correctly (we can use enquiry to search enquiry):

T24

```
ENQUIRY L L
```

Selection criteria

```
0 LK %...
1 EQ "4,19"
```

"%ABBREVIATION" is among them; we'll surely see the correct screen typing ABBREVIATION L:

T24

```
 Model Bank R14

    Abbreviation                        Original Text
 ------------------------------------------------------------------------------
 1  AAA                                 AA.ARRANGEMENT.ACTIVITY,
 2  AB                                  ABBREVIATION
 3  ABB                                 ABBREVIATION
 4  AC                                  ACCOUNT
 5  ACAP                                ACCT.CAPITALISATION
 6  ACC                                 ACCOUNT
 7  ACCBAL                              LMM.ACCOUNT.BALANCES
 8  ACI                                 ACCOUNT.CREDIT.INT
 9  ADI                                 ACCOUNT.DEBIT.INT
10  AGC                                 ACCT.GEN.CONDITION
11  AGPC                                ACCT.GROUP.CONDITION
12  AMSH                                AM.SEGMENTED.HIERARCHY
13  AP                                  ACCOUNT.PARAMETER
14  APP                                 AZ.PRODUCT.PARAMETER
15  AT                                  ASSET.TYPE
16  AUTO                                EBS.AUTO.FUNCTION V HELPTEXT.MENU
 ------------------------------------------------------------------------------
 15 DEC 2014 22:27:09  USER (22 APR) INPUTTER          [23491,IPAGE   1 >>>3>>>
 ACTION _
 AWAITING PAGE INSTRUCTIONS
```

*(For Browser this setting doesn't matter at all.)*

In %ACCOUNT enquiry record we can also see why we had "BY CATEGORY BY CURRENCY" added to select earlier:

T24 application ENQUIRY

```
  4. 1 FIXED.SORT..... CATEGORY
  4. 2 FIXED.SORT..... CURRENCY
```

*(Besides fixed sort, fixed selection can also be assigned.)*

## 1.50 First T24 enquiry

- Chosen name: TRAIN.ENQ (unlike VERSION, no rules like starting from application @ID etc).
- Chosen application: USER.
- Chosen file: LIVE.
- Fixed selection: only active users.
- Fixed sort: USER.NAME.
- Fields to show: @ID, USER.NAME, COMPANY.CODE, START.DATE.PROFILE, END.DATE.PROFILE.

Initial screen:

T24

```
Model Bank R14          ENQUIRY, INPUT

    ENQUIRY........... TRAIN.ENQ
--------------------------------------------------------------------------------
  1 PAGE.SIZE ........ 4,19
  2 FILE.NAME......... USER
  3. 1 FIXE
  4. 1 FIXED.SORT.....
  5. 1 OPEN.BRACKET...
  6. 1 SELECTION.FLDS.
  7. 1. 1 GB SEL.LABEL
  8. 1 SEL.FLD.OPER...
  9. 1 REQUIRED.SEL...
 10. 1 CLOSE.BRACKET..
 11. 1 REL.NEXT.FIELD.
 12. 1 BUILD.ROUTINE..
 13. 1. 1 HEADER......
 14. 1 FIELD.NAME.....
 15. 1. 1 OPERATION...
 16. 1 COLUMN.........
--------------------------------------------------------------------------------
16 DEC 2014 00:00:51  USER (22 APR) INPUTTER          [8736,INPAGE 1   >>>6>>>
ACTION
```

Field 3 is really called FIXED.SELECTION but the name isn’t fully visible. In older times there was a limitation for field length (54) due to screen width limit. Field name limit was 18 characters for single-valued fields, 15 for multi-valued and 12 for sub-valued ones so field number, name and contents would fit the screen.

We can see it in ABBREVIATION:

T24

```
Model Bank R14          Abbreviation INPUT

    ABBREVIATION.CODE. WWWWW
------------------------------------------------------------------------------
  1 ORIGINAL.TEXT..... _                                                     |
  2 RECORD.STATUS.....
  3 CURR.NO...........
  4. 1 INPUTTER.......
  5. 1 DATE.TIME......
  6 AUTHORISER........
  7 CO.CODE...........
  8 DEPT.CODE.........
  9 AUDITOR.CODE......
 10 AUDIT.DATE.TIME...







------------------------------------------------------------------------------
30 MAR 2015 15:36:32  USER (22 APR) VLADIMIR.K        [7755,INPAGE 1
ACTION
```

When we edit the field ORIGINAL.TEXT which has maximum length 54 we see that its right side indicated by pipe symbol is exactly under “ruler” above it.  Length of “ORIGINAL.TEXT.....”  is 18, dots are for vertical aligning.  In case of a multi-valued field (INPUTTER) 3 positions are eaten by value number (“ 1 ” after “4.”), hence the limit of 15 instead of 18.

Back to ENQUIRY. “7.1.1 GB SEL.LABEL” is so-called “language” field where sub-values level is used for input in different languages.  If we go to field 7.1.1 and type ( Enter the second language subvalue will open:

T24

```
  7. 1. 1 GB SEL.LABEL
  7. 1. 2 FR SEL.LABEL
```

*(It can be removed, similarly to multi-value, inputting dash in it.)*

See also LANGUAGE application:

T24

```
Model Bank R14           LANGUAGE - Default List

   ID   MNEMONIC  DESCRIPTION                          AMOUNT.FORMAT    FUNCT.
------------------------------------------------------------------------------
1    1  GB        English
2    2  FR        French
3    3  DE        German
4    4  ES        Spanish
```

Later length limitation was no more considered as a rule so - pressing F7 on field 3 - we can estimate the pipe position indicating field length that really is 80:

jsh

```
CT DICT F.ENQUIRY FIXED.SELECTION
```

Output

```
   FIXED.SELECTION
001 D
002 3
003
004 FIXED.SELECTION
005 80L                          <=====
006 M
```

In the search screen all field names are shown completely (ENQUIRY L L):

T24

```
Model Bank R14          ENQUIRY LIST


 ------------------------------------------------------------------------------
       SELECTION FIELDS - LIVE FILE      [EQ NE LT GT LE GE RG NR LK UL]
 ------------------------------------------------------------------------------
 0.. @ID                   0A. ENQUIRY               1.. PAGE.SIZE
 2.. FILE.NAME             3.. FIXED.SELECTION       4.. FIXED.SORT
 5.. OPEN.BRACKET          6.. SELECTION.FLDS        7.. SEL.LABEL
...
```

First iteraton of enquiry setup:

T24

```
    ENQUIRY.......... TRAIN.ENQ
--------------------------------------------------------------------------------
  1 PAGE.SIZE ........ 4,19
  2 FILE.NAME......... USER
  3. 1 FIXE END.DATE.PROFILE GE !TODAY
  4. 1 FIXED.SORT..... USER.NAME
 13. 1. 1 HEADER...... ACTIVE USERS
 14. 1 FIELD.NAME..... @ID
 15. 1. 1 OPERATION... @ID
 16. 1 COLUMN......... 1
 17. 1 LENGTH.MASK.... 16L
 35. 1 SINGLE.MULTI... S
 14. 2 FIELD.NAME..... USER.NAME
 15. 2. 1 OPERATION... USER.NAME
 16. 2 COLUMN......... 18
 17. 2 LENGTH.MASK.... 15L
 35. 2 SINGLE.MULTI... S
 14. 3 FIELD.NAME..... COMPANY.CODE
 15. 3. 1 OPERATION... COMPANY.CODE
 16. 3 COLUMN......... 35
 17. 3 LENGTH.MASK.... 11L
 35. 3 SINGLE.MULTI... M
 14. 4 FIELD.NAME..... START.DATE.PROFILE
 15. 4. 1 OPERATION... START.DATE.PROFILE
 16. 4 COLUMN......... 47
 17. 4 LENGTH.MASK.... 11R
 35. 4 SINGLE.MULTI... S
 14. 5 FIELD.NAME..... END.DATE.PROFILE
 15. 5. 1 OPERATION... END.DATE.PROFILE
 16. 5 COLUMN......... 59
 17. 5 LENGTH.MASK.... 11R
 35. 5 SINGLE.MULTI... S
--------------------------------------------------------------------------------
```

*(COLUMN settings are necessary for terminal mode only, for Browser there can be anything - like 1, 2 etc.  If there's nothing in COLUMN field, corresponding enquiry field won't be displayed - it's used for technical fields like ones for intermediate calculations.)*

In "END.DATE.PROFILE GE !TODAY" "!TODAY" is current business day (which is usually but not necessarily the same as calendar day).  See enquiry output (without selection):

T24 "AWAITING APPLICATION" prompt

```
QUERY TRAIN.ENQ
```

Result

```
Model Bank R14


--------------------------------------------------------------------------------
ACCTEXEC         Account Execut|  GB0010001      20140515    20160924
AUTHORISER       AUTHORISER       GB0010001      20140328    20200328
                                  GB0010002
                                  EU0010001
                                  SG0010001
                                  GB0010005
                                  GB0010003
                                  GB0010004
BUILDUSER279     AUTHORISER       GB0010001      20090923    20151230
GOPIRAMK         AUTHORISER       GB0010001      20091106    20151230
AUTO.MC          AUTOMC           GB0010001      20131022    20201231
BASEAUTH         BAUTH            GB0010001      20110207    20201231
                                  GB0010002
                                  EU0010001
                                  SG0010001
                                  GB0010005
--------------------------------------------------------------------------------
16 DEC 2014 00:47:45  USER (22 APR) INPUTTER          [21299,IPAGE   1 >>>3>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

Column 2 has display format less than field length so the pipe acts as a terminator.

It doesn’t matter in Browser:

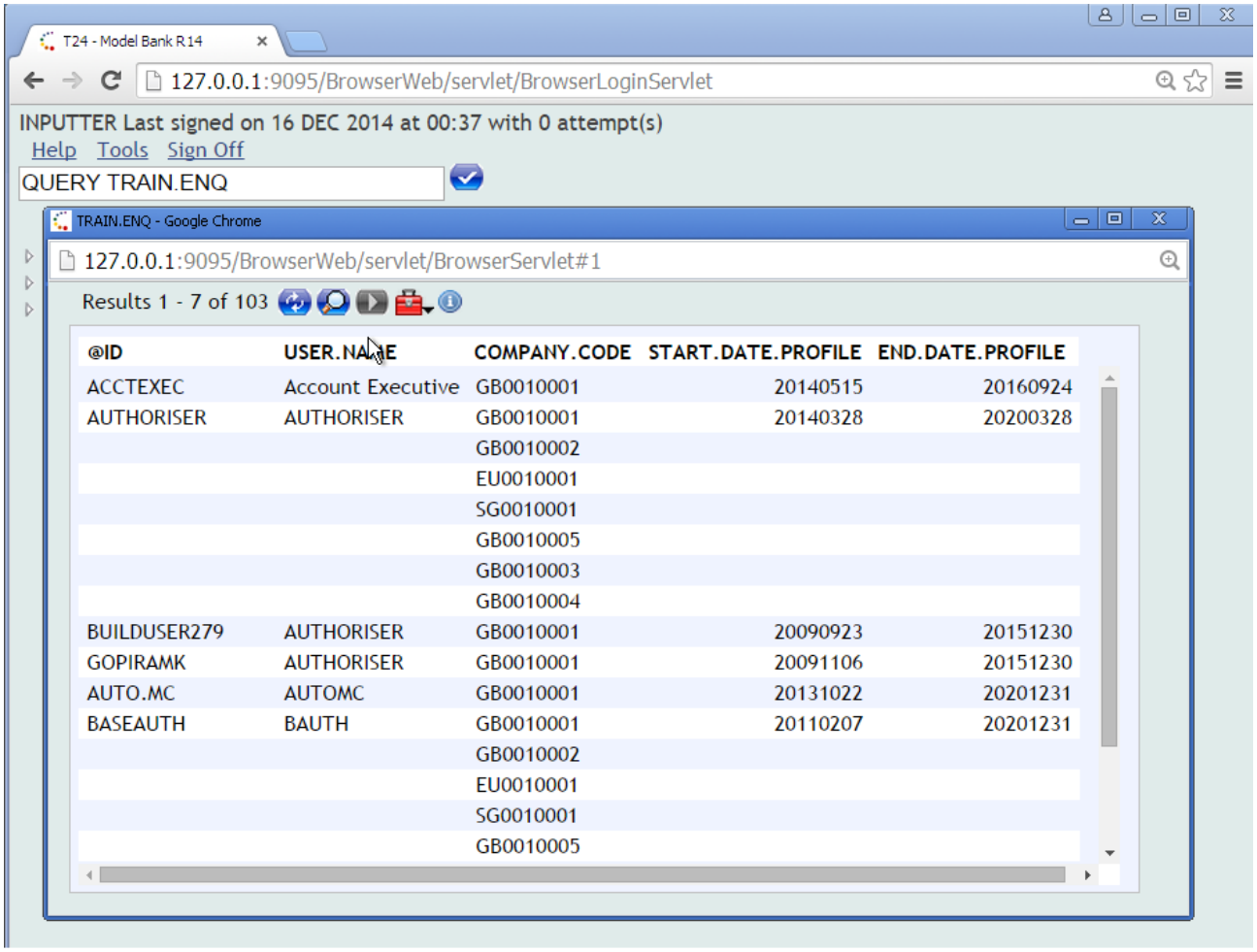


**Figure 29:  First enquiry in Browser.**

If we save it to csv, the resulting file will look as:

TRAIN.ENQ.csv

```
."@ID","USER.NAME","COMPANY.CODE","START.DATE.PROFILE","END.DATE.PROFILE"
-
"ACCTEXEC","Account Executive","GB0010001","20140515","20160924"
"AUTHORISER","AUTHORISER","GB0010001","20140328","20200328"
"","","GB0010002","",""
"","","EU0010001","",""
"","","SG0010001","",""
"","","GB0010005","",""
"","","GB0010003","",""
"","","GB0010004","",""
"BUILDUSER279","AUTHORISER","GB0010001","20090923","20151230"
"GOPIRAMK","AUTHORISER","GB0010001","20091106","20151230"
...
```

*(COMPANY.CODE is multi-valued so it might occupy more than one line.)*

Enquiry with selection - see which records were copied without changing user name AUTHORISER:

T24

```
ENQ TRAIN.ENQ
```

T24 enquiry selection screen

```
Model Bank R14          ENQUIRY INPUT

    SELECT NAME....... TRAIN.ENQ
--------------------------------------------------------------------------------
  1. 1 SELECTION.TYPE. <k>
  2. 1 SELECTION.FIELD @ID
  3. 1 OPERAND........
  4. 1. 1 LIST........
  1. 2 SELECTION.TYPE. <k>
  2. 2 SELECTION.FIELD USER.ID
  3. 2 OPERAND........
  4. 2. 1 LIST........
  1. 3 SELECTION.TYPE.
  2. 3 SELECTION.FIELD USER.NAME                                          <=====
  3. 3 OPERAND........ EQ                                                 <=====
  4. 3. 1 LIST........ AUTHORISER                                         <=====
  1. 4 SELECTION.TYPE.
  2. 4 SELECTION.FIELD SIGN.ON.NAME
  3. 4 OPERAND........
  4. 4. 1 LIST........
-- LAST SIGN.ON, DATE: 16 DEC 2014     TIME: 00:50        ATTEMPTS: 0 --------
16 DEC 2014 01:04:39  USER (22 APR) INPUTTER          [59,IN] PAGE 1   >>25>>>
ACTION
```

T24 enquiry results

```
Model Bank R14


--------------------------------------------------------------------------------
AUTHORISER       AUTHORISER       GB0010001      20140328    20200328
                                  GB0010002
                                  EU0010001
                                  SG0010001
                                  GB0010005
                                  GB0010003
                                  GB0010004
BUILDUSER279     AUTHORISER       GB0010001      20090923    20151230
GOPIRAMK         AUTHORISER       GB0010001      20091106    20151230









--------------------------------------------------------------------------------
```

User selection is remembered in:

jsh

```
CT F.ENQUIRY.SELECT 'TRAIN.ENQ_GB0010001_INPUTTER'
```

Output

```
    TRAIN.ENQ_GB0010001_INPUTTER
001 USER.NAME]EQ]AUTHORISER]
```

Selection process in Browser:

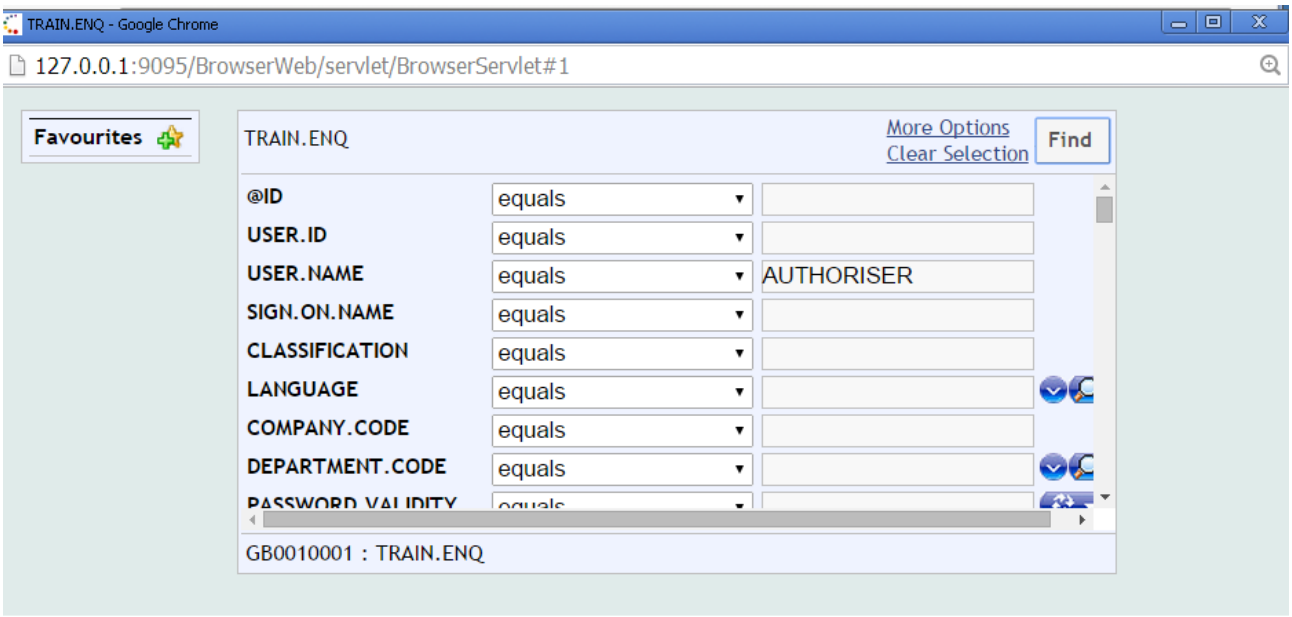


**Figure 30:  Enquiry selection in Browser.**

Other comparison options are:

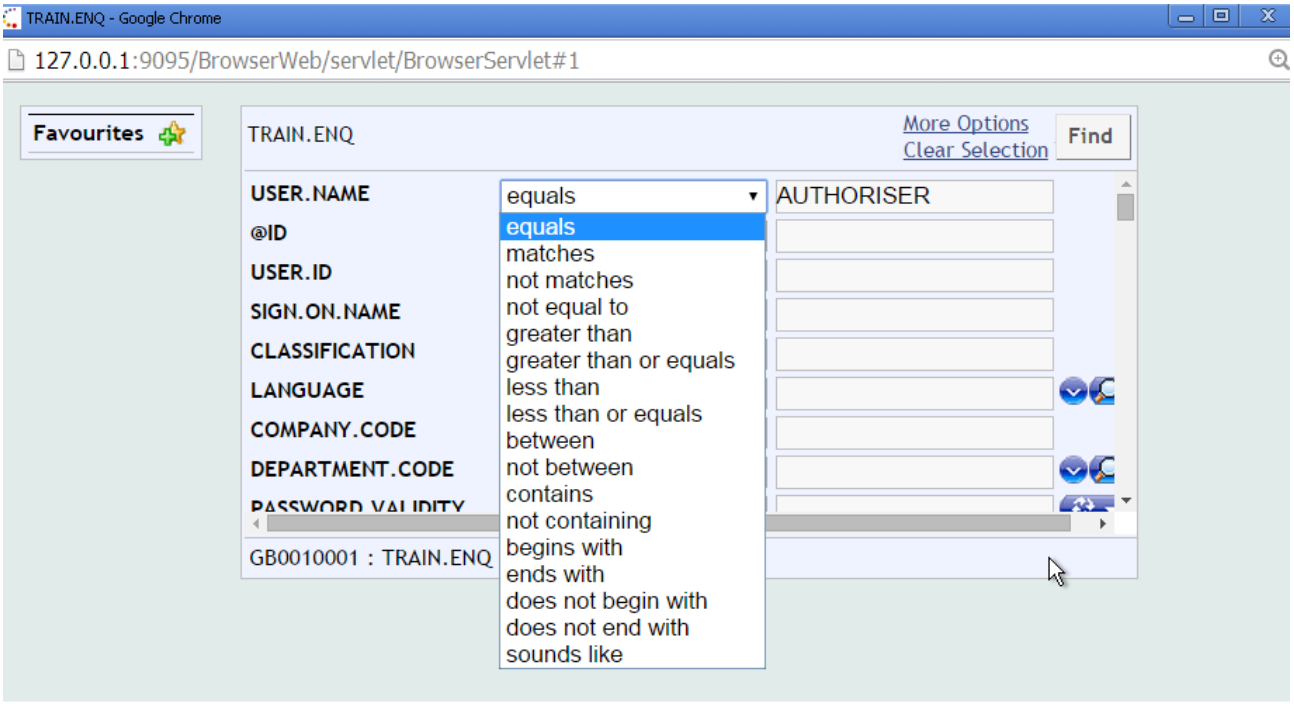


**Figure 31: Comparison options in Browser.**

Clicking to plus icon near “Favourites” raises a dialog:

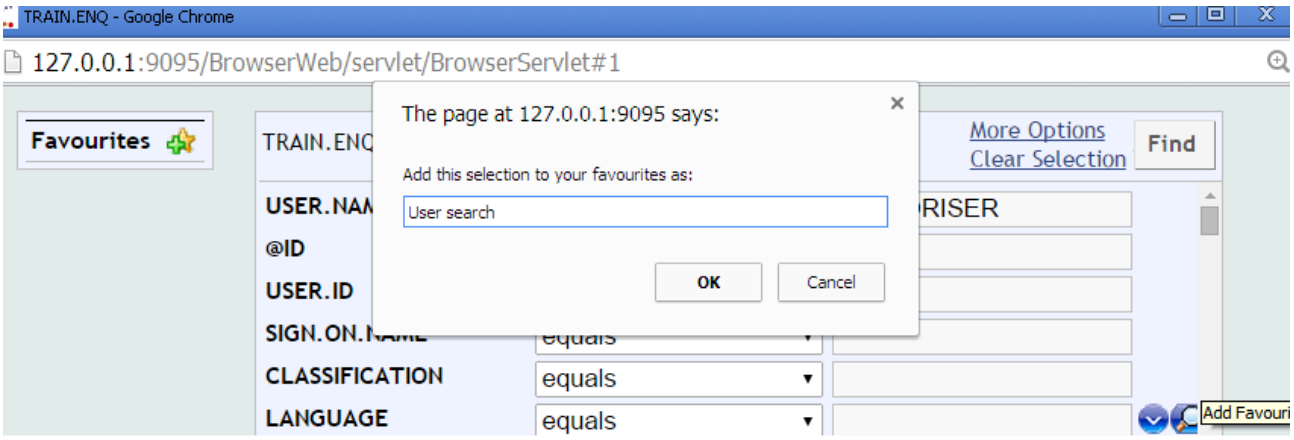


**Figure 32: Adding to favourites.**

Next time in this enquiry there's "user search":

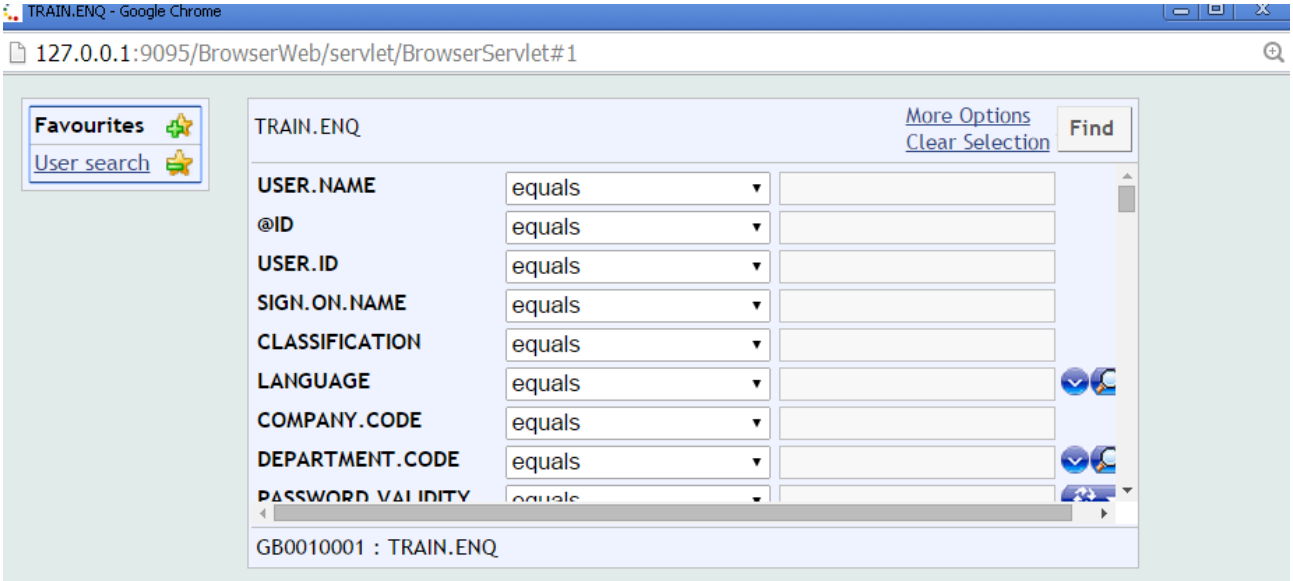


**Figure 33: Favourite added.**

Again, this option is saved to file F.ENQUIRY.SELECT. See how many records corresponding to this enquiry we now have, some of them are probably redundant:

jsh

```
LIST F.ENQUIRY.SELECT LIKE TRAIN.ENQ...INPUTTER...
```

Output

```
TRAIN.ENQ_GB0010001_INPUTTER_1418693681602_SELECTION
TRAIN.ENQ-INPUTTER
TRAIN.ENQ_GB0010001_INPUTTER_1418693681602_
TRAIN.ENQ_GB0010001_INPUTTER_1418691026126_
TRAIN.ENQ_GB0010001_INPUTTER_1418693719568_
TRAIN.ENQ_GB0010001_INPUTTER
TRAIN.ENQ_GB0010001_INPUTTER_1418692136127_SELECTION
TRAIN.ENQ_GB0010001_INPUTTER_1418693719568_SELECTION
TRAIN.ENQ_GB0010001_INPUTTER_1418692136127_
TRAIN.ENQ_GB0010001_INPUTTER_1418692767967_
TRAIN.ENQ_GB0010001_INPUTTER_1418692951889_SELECTION
TRAIN.ENQ_GB0010001_INPUTTER_1418691026126_SELECTION
TRAIN.ENQ_GB0010001_INPUTTER_1418692951889_
TRAIN.ENQ_GB0010001_INPUTTER_1418692767967_SELECTION
```

## 1.51 USER.ABBREVIATION

But when in Browser you click “Tools - My profile - Organise favourites” - the record of another T24 application will be opened - USER.ABBREVIATION:

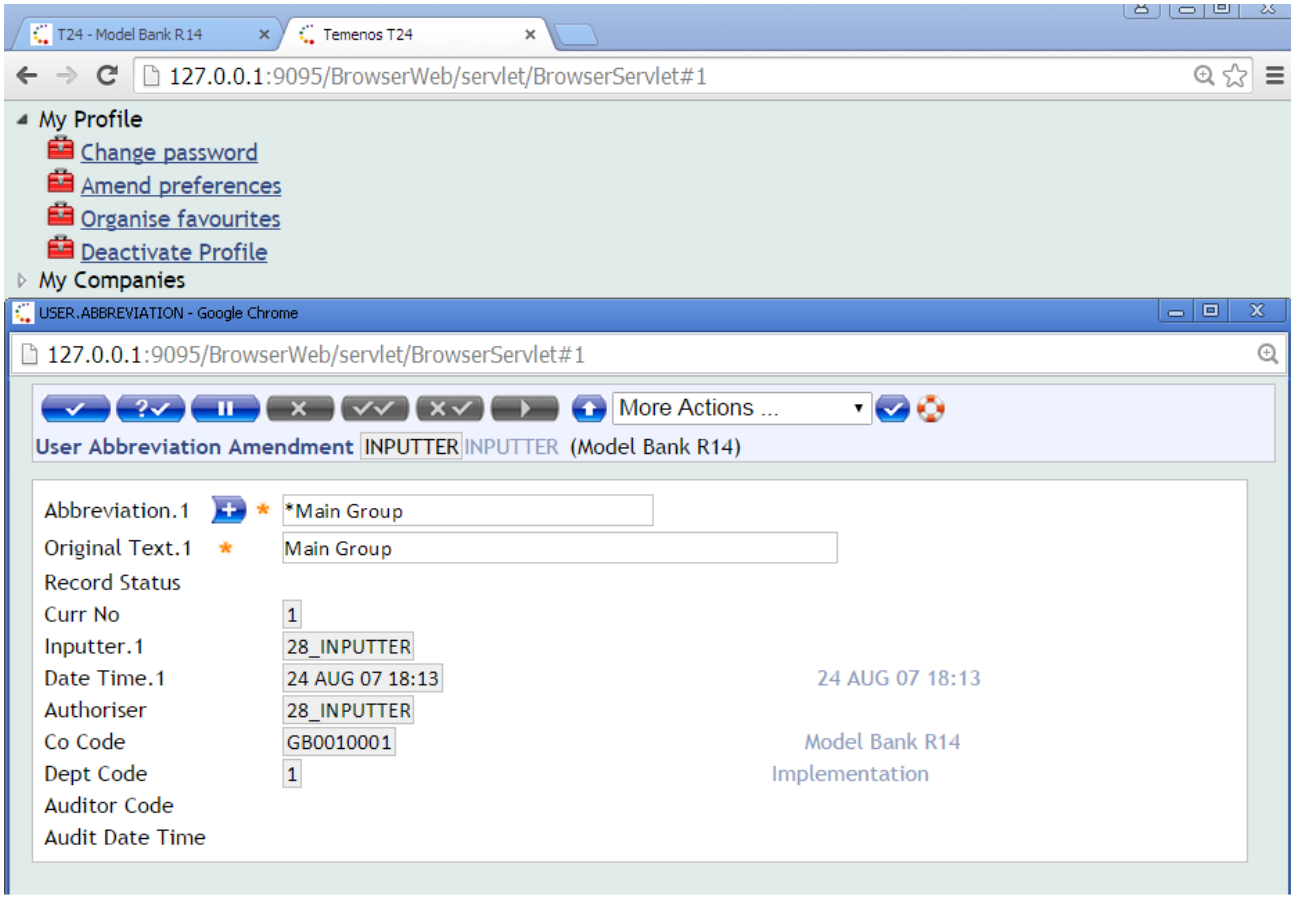


**Figure 34: User abbreviations.**

Expand the multi-value fields association and add new item:

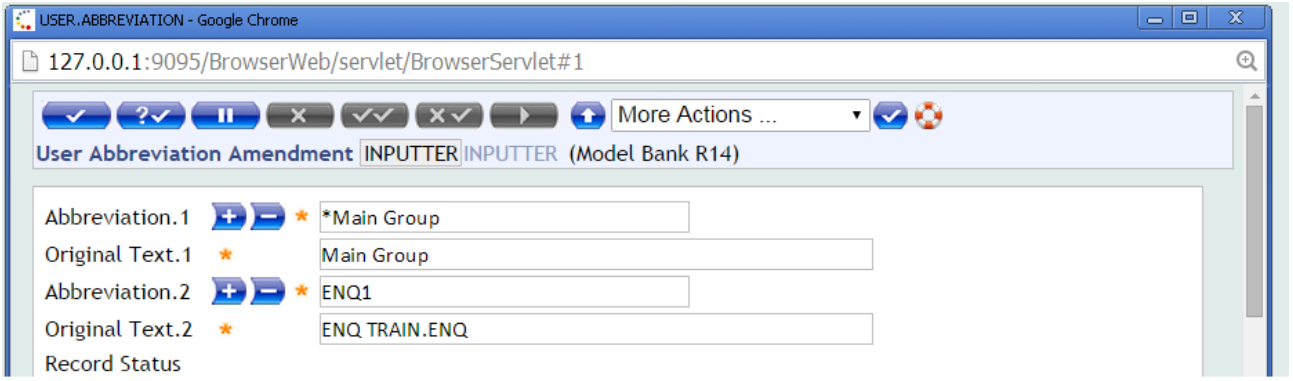


**Figure 35: Adding an item to user abbreviations.**

Commit the record.  The message appears:

T24 Browser

```
Txn Complete: INPUTTER 01:53:08 16 DEC 2014 USER.ABBREVIATION,AMEND I
```

USER.ABBREVIATION,AMEND is the VERSION that was launched for this task.  See how many authorisations it has:

jsh

```
LIST F.VERSION 'USER.ABBREVIATION,AMEND' NO.OF.AUTH
```

The result is 0 so we are supposed to use “ENQ1” straight away both on Browser command line and in T24 “AWAITING APPLICATION” prompt.  However, it looks like it no more works in Browser - it gives the error “ENQUIRY.SELECTION application not supported in Browser” though in terminal all is OK...  If we add the user abbreviation to launch “SPF S SYSTEM”, terminal mode is OK again while Browser just opens SPF application without opening the record.

## 1.52 More OFS

Let’s use OFS to populate the above-mentioned record of USER.ABBREVIATION. We’ll add and delete values in multi-value association of fields ABBREVIATION and ORIGINAL.TEXT via tSS TAABS command.

Add new set:

OFS input

```
USER.ABBREVIATION,/I/PROCESS//0,INPUTT/123456,INPUTTER,ABBREVIATION:-1:=MYSPF,
   ORIGINAL.TEXT:-1:="SPF S SYSTEM"
```

Result:

T24

```
 Model Bank R14          USER.ABBREVIATION SEE

     USER.ID........... INPUTTER            INPUTTER
 ------------------------------------------------------------------------------
   1. 1 ABBREVIATION... *Main Group
   2. 1 ORIGINAL.TEXT.. Main Group
   1. 2 ABBREVIATION... ENQ1
   2. 2 ORIGINAL.TEXT.. ENQ TRAIN.ENQ
   1. 3 ABBREVIATION... MYSPF
   2. 3 ORIGINAL.TEXT.. SPF S SYSTEM
   ...
```

If we repeat that OFS message again we'll see the error message:

OFS output

```
INPUTTER//-1/NO,ABBREVIATION:4:1=DUPLICATE
```

Let's try to add another set populating only one field:

OFS input

```
USER.ABBREVIATION,/I/PROCESS//0,INPUTT/123456,INPUTTER,ABBREVIATION:-1:=MYCOMPANY
```

OFS output

```
INPUTTER//-1/NO,ORIGINAL.TEXT:4:1=INPUT MISSING
```

...meaning that the field ORIGINAL.TEXT is mandatory.

Add another set:

OFS input

```
USER.ABBREVIATION,/I/PROCESS//0,INPUTT/123456,INPUTTER,ABBREVIATION:-1:=MYCOMPANY,
    ORIGINAL.TEXT::="COMPANY S BNK"
```

OFS output

```
INPUTTER//-1/NO,ORIGINAL.TEXT:2:1=INPUT MISSING,ORIGINAL.TEXT:3:1=INPUT MISSING,
   ORIGINAL.TEXT:4:1=INPUT MISSING
```

We accidentally tried to replace the whole field ORIGINAL.TEXT with a single value so the core assumed values #2, #3 and #4 being empty.

Finally:

OFS input

```
USER.ABBREVIATION,/I/PROCESS//0,INPUTT/123456,INPUTTER,ABBREVIATION:-1:=MYCOMPANY,
    ORIGINAL.TEXT:-1:="COMPANY S BNK"
```

Resulting record:

T24

```
    USER.ID........... INPUTTER            INPUTTER
 ------------------------------------------------------------------------------
   1. 1 ABBREVIATION... *Main Group
   2. 1 ORIGINAL.TEXT.. Main Group
   1. 2 ABBREVIATION... ENQ1
   2. 2 ORIGINAL.TEXT.. ENQ TRAIN.ENQ
   1. 3 ABBREVIATION... MYSPF
   2. 3 ORIGINAL.TEXT.. SPF S SYSTEM
   1. 4 ABBREVIATION... MYCOMPANY
   2. 4 ORIGINAL.TEXT.. COMPANY S BNK
   4 CURR.NO........... 10
   5. 1 INPUTTER....... 24937_INPUTTER__OFS_TAABS
   ...
```

To clear the contents of particular fields is generally possible (setting content to NULL) but not in this case:

OFS input

```
USER.ABBREVIATION,/I/PROCESS//0,INPUTT/123456,INPUTTER,ABBREVIATION:3:=NULL,
    ORIGINAL.TEXT:3:=NULL
```

OFS output

```
INPUTTER//-1/NO,ABBREVIATION:3:1=INPUT MISSING
```

Without knowing the contents of the record it’s a bit hard to remove, e.g., the set #3 should we want that.  VALIDATE option can help to retrieve current values; OFS is also capable to launch an enquiry:

OFS input

```
ENQUIRY.SELECT,,INPUTT/123456,%USER.ABBREVIATION,@ID:EQ=INPUTTER
```

OFS output

```
,@ID::@ID/ABBREVIATION::ABBREVIATION/RECORD.STATUS::RECORD.STATUS,
"INPUTTER        "    "*Main Group ENQ1 MYSPF MYCOMPANY   "   "             "
```

As we see, “percent” enquiry doesn’t give us all data - output is the same as we will see after T24 command USER.ABBREVIATION L:

T24

```
   ID                ABBREVIATION                          RECORD.STATUS FUNCT.
 ------------------------------------------------------------------------------
 1  AUTHORISER        *Main Group
 2  INPUTTER          *Main Group ENQ1 MYSPF MYCOMPANY
```

OFS replace option can help if we know what has to be retained (sets 2 and 4 in the following example):

OFS input

```
USER.ABBREVIATION,/I/PROCESS//0,INPUTT/123456////1,INPUTTER,ABBREVIATION:1:=ENQ1,
                                                  ^^^^^
    ORIGINAL.TEXT:1:="ENQ TRAIN.ENQ",ABBREVIATION:2:=MYCOMPANY,
    ORIGINAL.TEXT:2:="COMPANY S BNK"
```

Resulting record:

T24

```
    USER.ID.......... INPUTTER            INPUTTER
 ------------------------------------------------------------------------------
   1. 1 ABBREVIATION... ENQ1
   2. 1 ORIGINAL.TEXT.. ENQ TRAIN.ENQ
   1. 2 ABBREVIATION... MYCOMPANY
   2. 2 ORIGINAL.TEXT.. COMPANY S BNK
   4 CURR.NO........... 11
```

## 1.53 Enhancing the enquiry

We’ll assign meaningful column headers, correct dates output and add another column - work days left till user profile ends (using enquiry “conversion routine”):

jBC

```
SUBROUTINE SUBR.CONV

$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_ENQUIRY.COMMON

   end_date = O.DATA             ;* incoming/outgoing data
                                 ;* declared in I_ENQUIRY.COMMON
   the_diff = 'W'
   CALL CDD('', TODAY, end_date, the_diff)
   O.DATA = the_diff

   RETURN
END
```

Changes to ENQUIRY record:

T24

```
    ENQUIRY........... TRAIN.ENQ
------------------------------------------------------------------------------
 ...
 22. 1. 1 GB FIELD.LBL T24 ID
 ...
 22. 2. 1 GB FIELD.LBL User name
 ...
 22. 3. 1 GB FIELD.LBL Branch
 ...
 18. 4. 1 C ICONV D
 18. 4. 2 C OCONV D
 ...
 22. 4. 1 GB FIELD.LBL Active from
 ...
 18. 5. 1 C ICONV D
 18. 5. 2 C OCONV D2
 ...
 22. 5. 1 GB FIELD.LBL Valid until
 ...
```

New field:

T24

```
    ENQUIRY........... TRAIN.ENQ
------------------------------------------------------------------------------
 ...
 14. 6 FIELD.NAME..... DIFF
 15. 6. 1 OPERATION... END.DATE.PROFILE
 16. 6 COLUMN......... 71
 17. 6 LENGTH.MASK.... 11R
 18. 6. 1 C @ SUBR.CONV                              <=== routine is attached here
 22. 6. 1 GB FIELD.LBL Days to work
 35. 6 SINGLE.MULTI... S
```

Results:

T24

```
Model Bank R14

--------------------------------------------------------------------------------
ACCTEXEC         Account Execut|  GB0010001   15 MAY 2014   24 SEP 16         633
AUTHORISER       AUTHORISER       GB0010001   28 MAR 2014   28 MAR 20        1548
                                  GB0010002
                                  EU0010001
                                  SG0010001
                                  GB0010005
                                  GB0010003
                                  GB0010004
BUILDUSER279     AUTHORISER       GB0010001   23 SEP 2009   30 DEC 15         441
GOPIRAMK         AUTHORISER       GB0010001   06 NOV 2009   30 DEC 15         441
AUTO.MC          AUTOMC           GB0010001   22 OCT 2013   31 DEC 20        1747
BASEAUTH         BAUTH            GB0010001   07 FEB 2011   31 DEC 20        1747
                                  GB0010002
                                  EU0010001
                                  SG0010001
                                  GB0010005
--------------------------------------------------------------------------------
```

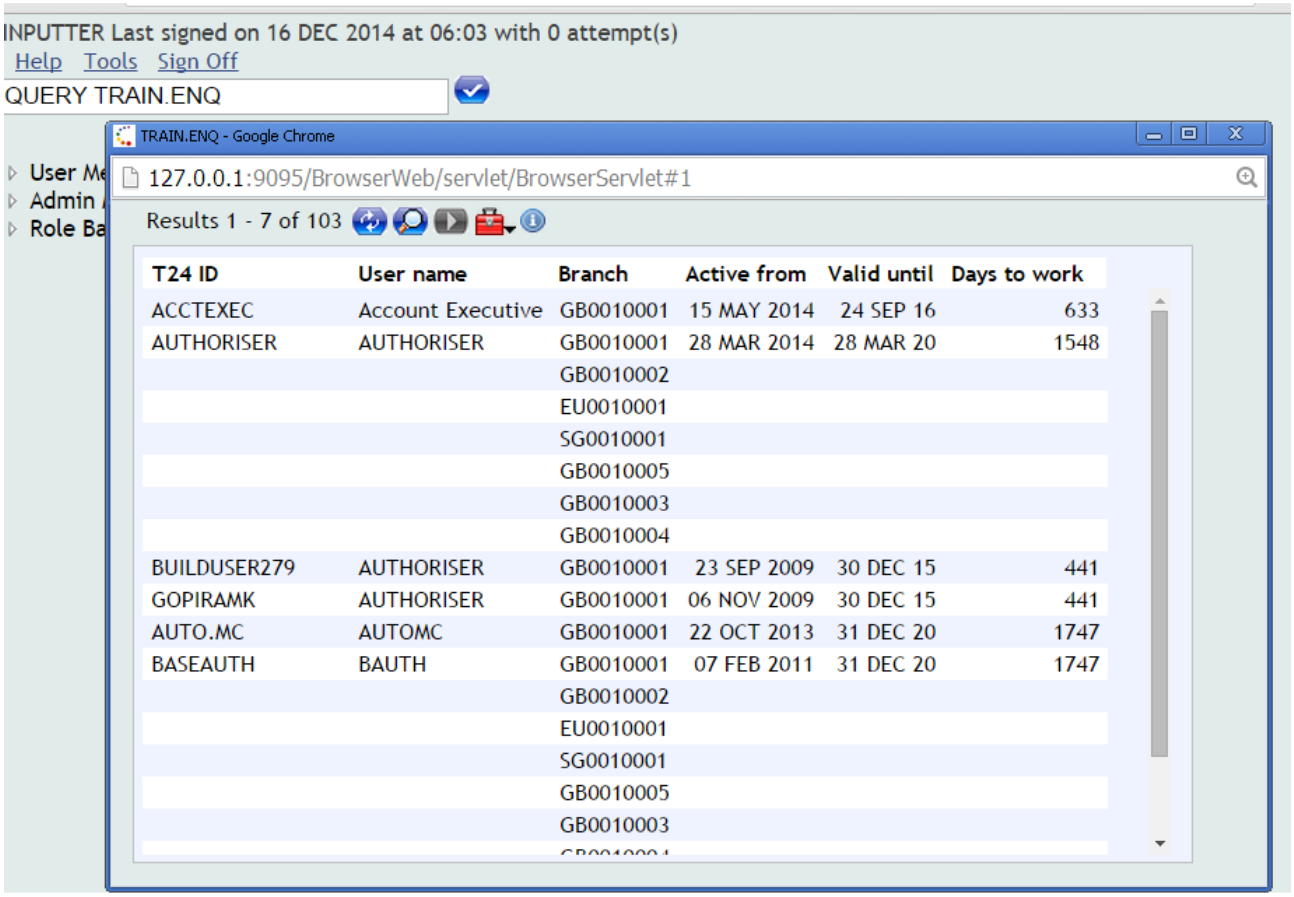


**Figure 36: Enhanced enquiry in Browser.**

*(2-digit year in the column “Valid until” is stipulated by conversion code “D2” rather than “D” in ENQUIRY field 18.5.2.)*

*(If a new routine is compiled or existing one is amended, jbase agent restart is required for Browser to recognize changes.)*

To be able to go directly to each record we need to add something to ENQUIRY record:

T24 enquiry record

```
 36. 1 ENQU USER S @ID
 38. 1 LABEL.FIELD.... @ID 1
 39. 1. 1 GB NXT.DESC. See user profile
```

*(Field 36 is “ENQUIRY.NAME”.)*

Also, to see numbers at the leftmost column in Classic mode we need to move all columns to the right.  Now try it:

T24

```
Model Bank R14


--------------------------------------------------------------------------------
1  ACCTEXEC         Account Execut|  GB0010001   15 MAY 2014   24 SEP 16         633
2  AUTHORISER       AUTHORISER       GB0010001   28 MAR 2014   28 MAR 20        1548
                                     GB0010002
                                     EU0010001
                                     SG0010001
                                     GB0010005
                                     GB0010003
                                     GB0010004
3  BUILDUSER279     AUTHORISER       GB0010001   23 SEP 2009   30 DEC 15         441
4  GOPIRAMK         AUTHORISER       GB0010001   06 NOV 2009   30 DEC 15         441
5  AUTO.MC          AUTOMC           GB0010001   22 OCT 2013   31 DEC 20        1747
6  BASEAUTH         BAUTH            GB0010001   07 FEB 2011   31 DEC 20        1747
                                     GB0010002
                                     EU0010001
                                     SG0010001
                                     GB0010005
--------------------------------------------------------------------------------
16 DEC 2014 09:08:02  USER (22 APR) INPUTTER          [26586,IPAGE   1 >>>3>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

Now - 1 Enter I Enter F5:

T24

```
ACTION I                                    INVALID FUNCTION IN ENQUIRY MODE
AWAITING FUNCTION
```

All right, back (F1), S function:

T24

```
Model Bank R14          USER PROFILE

    USER.ID........... ACCTEXEC
------------------------------------------------------------------------------
  1 USER.NAME......... Account Executive
  2 SIGN.ON.NAME...... ACCTEXEC1
  3 CLASSIFICATION.... INT
  4 LANGUAGE.......... 1                   English
  5. 1 COMPANY.CODE... GB0010001           Model Bank R14
  6 DEPARTMENT.CODE... 1                   Implementation
  7 PASSWORD.VALIDITY. 01 NOV 2014 M0601   01 NOV 2014 Every 6 months on day 1
  8 START.DATE.PROFILE 15 MAY 2014
  9 END.DATE.PROFILE.. 24 SEP 2016
 10. 1 START.TIME..... 00:00
 11. 1 END.TIME....... 24:00
 12 TIME.OUT.MINUTES.. 999
 13 ATTEMPTS.......... 9
 17. 1 COMPANY.RESTR.. GB0010001           Model Bank R14
 18. 1 APPLICATION.... ALL.PG
 20. 1 FUNCTION....... A 2 B C D E F H I L P R S V
------------------------------------------------------------------------------
16 DEC 2014 09:33:17  USER (22 APR) INPUTTER          [10650,IPAGE 1   >>>2>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

And in Browser:

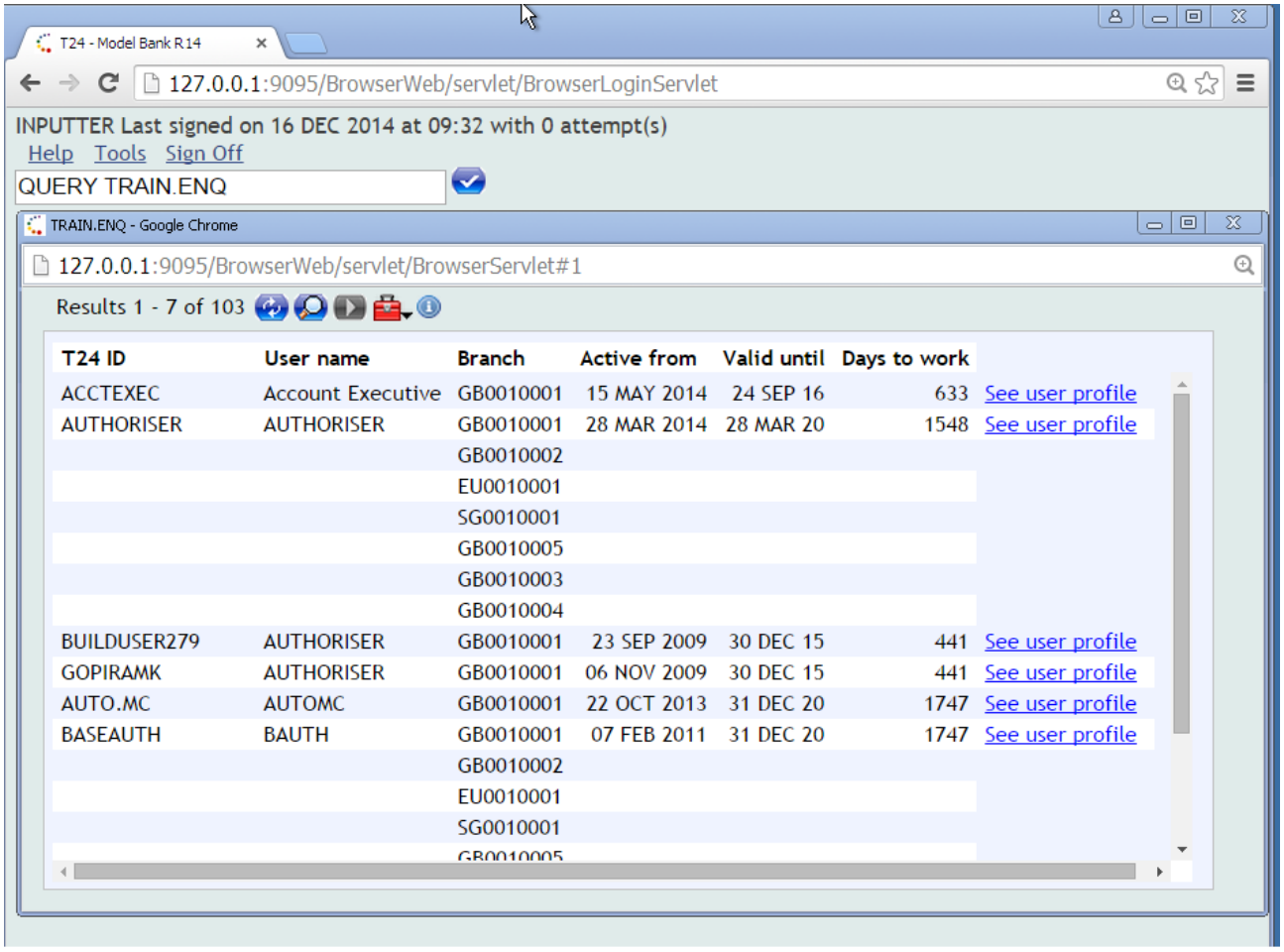


**Figure 37: Go-to-record option in enquiry.**

Click “See user profile”:

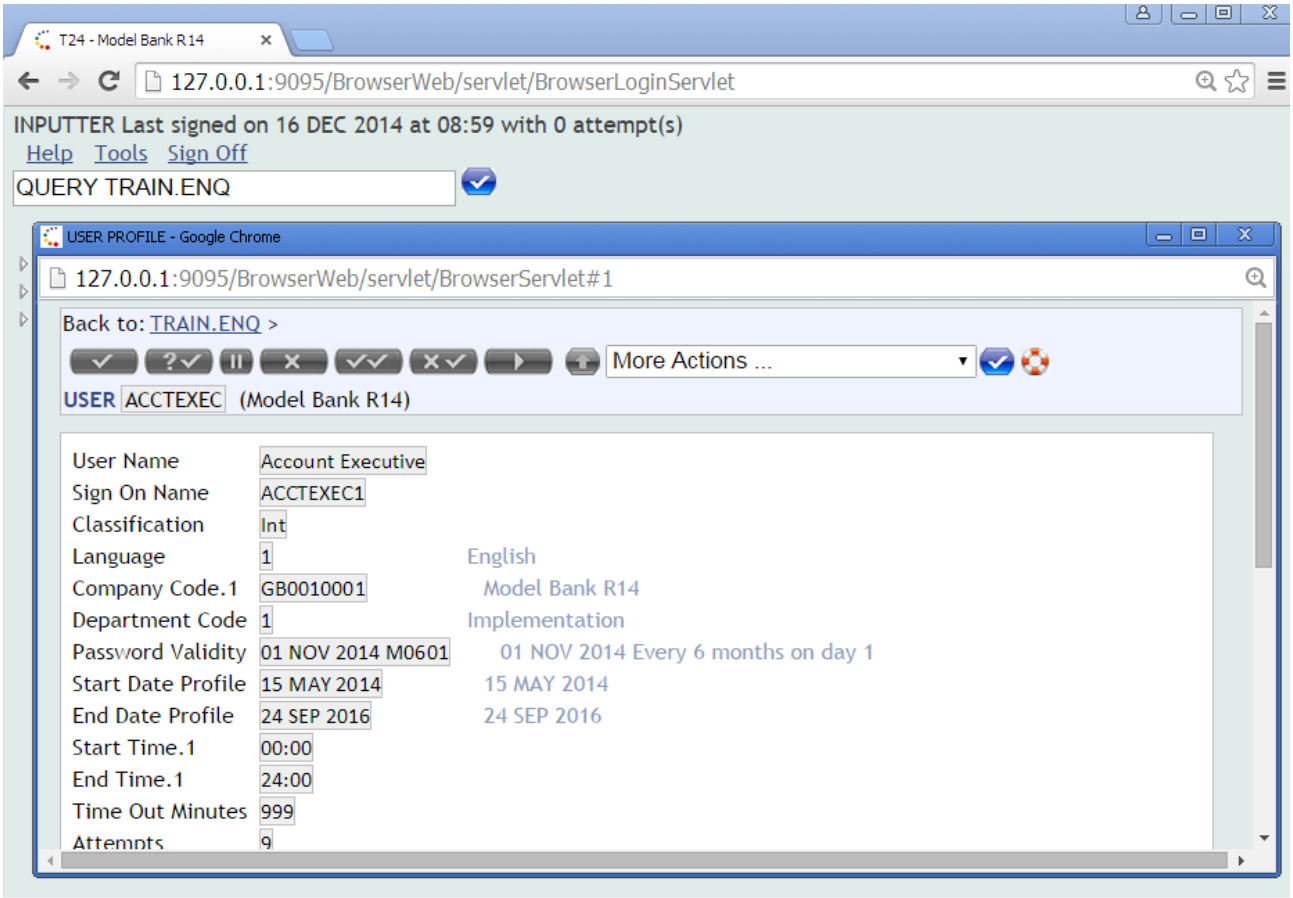


**Figure 38:  Record opened from enquiry.**

Finally, let’s output branch name instead of a code:

T24 - change to ENQUIRY record TRAIN.ENQ

```
14. 3 FIELD.NAME..... COMPANY.CODE
15. 3. 1 OPERATION... COMPANY.CODE
16. 3 COLUMN......... 38
17. 3 LENGTH.MASK.... 11L
18. 3. 1 C L COMPANY,COMPANY.NAME            <===== here ('L' stands for 'LINK')
```

Result:

T24

```
Model Bank R14


--------------------------------------------------------------------------------
1 ACCTEXEC          Account Execut|  Model Bank| 15 MAY 2014   24 SEP 16         633
2 AUTHORISER        AUTHORISER       Model Bank| 28 MAR 2014   28 MAR 20        1548
                                     MF LEAD CO|
                                     Model Bank|
                                     Model Bank|
                                     Model Bank|
                                     MF Branch 1
                                     MF Branch 2
3 BUILDUSER279      AUTHORISER       Model Bank| 23 SEP 2009   30 DEC 15         441
4 GOPIRAMK          AUTHORISER       Model Bank| 06 NOV 2009   30 DEC 15         441
5 AUTO.MC           AUTOMC           Model Bank| 22 OCT 2013   31 DEC 20        1747
6 BASEAUTH          BAUTH            Model Bank| 07 FEB 2011   31 DEC 20        1747
                                     MF LEAD CO|
                                     Model Bank|
                                     Model Bank|
                                     Model Bank|
--------------------------------------------------------------------------------
03 FEB 2015 15:33:54  USER (22 APR) TRAIN.01          [1806,INPAGE   1 >>>3>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

## 1.54 Save our work

For now we have several jBC source files and T24 application records developed (VERSION and ENQUIRY). To save it all for backup or for restore to another environment DL.DEFINE application is used (DL stands for “Data Library”).

This application has yet unknown to us type “W”:

jsh

```
LIST F.PGM.FILE 'DL.DEFINE' DESCRIPTION TYPE
```

Output

```
@ID......... DL.DEFINE
DESCRIPTION. File to hold the data library   definitions.
TYPE........ W
```

It means that this application doesn’t have either NAU or HIS file and the function “V” can be used to trigger a “special action”.

To save our work create a new DL.DEFINE record, e.g.  SAVE.20141219:

T24

```
Model Bank R14          DL.DEFINE INPUT

    UNIT.NAME......... TMNS000-SAVE.20141219
--------------------------------------------------------------------------------
  1. 1. 1 GB DESCRIPTN
  2. 1 GB SHORT.DESC..
  3 LANGUAGE/COUNTRY..
  4. 1 INDICES........
  5 OPERATION......... S
  6 SELECT.LIST.......
  7 TOP.LEVEL.TYPE....
  8 TOP.LEVEL.ITEM....
  9. 1 FILE.NAME......
 10. 1 RECO
 11. 1. 1 RECORD.DESC.
 12. 1 SAVED.FROM.....
 13. 1 SAVED.RELEASE..
 14. 1 SAVED.DATE.....
 15. 1 RESTORED.USER..
 16. 1 RESTD.COMPANY..
--------------------------------------------------------------------------------
16 DEC 2014 11:29:25  USER (22 APR) INPUTTER          [23158,IPAGE 1   >>>3>>>
ACTION
```

@ID was automatically prefixed with a cut from the license code contained in SPF record:

jsh

```
LIST F.SPF LICENSE.CODE
```

Output

```
@ID...    LICENSE.CODE...

SYSTEM    EURGBTMNS000
```

Back to DL.DEFINE. Populate the header; for each record that we need to save supply application and @ID.

Example:

T24

```
Model Bank R14          DL.DEFINE INPUT

    UNIT.NAME......... TMNS000-SAVE.20141219
------------------------------------------------------------------------------
  1. 1. 1 GB DESCRIPTN First save ever
  2. 1 GB SHORT.DESC.. SAVE 1
  3 LANGUAGE/COUNTRY..
  4. 1 INDICES........
  5 OPERATION......... S
  6 SELECT.LIST.......
  7 TOP.LEVEL.TYPE....
  8 TOP.LEVEL.ITEM....
  9. 1 FILE.NAME...... VERSION                                          <=====
 10. 1 RECO FUNDS.TRANSFER,TRAIN                                        <=====
            ^^^^^^^^^^^^^^^^^^^
 11. 1. 1 RECORD.DESC.
  9. 2 FILE.NAME...... ENQUIRY                                          <=====
 10. 2 RECO TRAIN.ENQ                                                   <=====
            ^^^^^^^^^
 11. 2. 1 RECORD.DESC.
 12. 1 SAVED.FROM.....
 13. 1 SAVED.RELEASE..
------------------------------------------------------------------------------
16 DEC 2014 11:50:22  USER (22 APR) INPUTTER          [24248,IPAGE 1   >>>3>>>
ACTION
```

*(Again field name is truncated due to big field size for field 10 - RECORD.NAME.)*

For subroutines put their directory (ETC.BP in our case) into field FILE.NAME and names - into field RECORD.NAME.

Enter everything and commit.  Record after saving:

T24

```
Model Bank R14          DL.DEFINE SEE

    UNIT.NAME......... TMNS000-SAVE.20141219
------------------------------------------------------------------------------
  1. 1. 1 GB DESCRIPTN First save ever
  2. 1 GB SHORT.DESC.. SAVE 1
  5 OPERATION......... S
  9. 1 FILE.NAME...... VERSION
 10. 1 RECO FUNDS.TRANSFER,TRAIN
  9. 2 FILE.NAME...... ENQUIRY
 10. 2 RECO TRAIN.ENQ
  9. 3 FILE.NAME...... VERSION
 10. 3 RECO ABBREVIATION,TRAIN
  9. 4 FILE.NAME...... PGM.FILE
 10. 4 RECO SUBR.ANC
  9. 5 FILE.NAME...... EB.API
 10. 5 RECO SUBR.VAL
  9. 6 FILE.NAME...... EB.API
 10. 6 RECO SUBR.INP
```

```
  9. 7 FILE.NAME...... ETC.BP
 10. 7 RECO PROG.0
  9. 8 FILE.NAME...... ETC.BP
 10. 8 RECO SUBR.0
  9. 9 FILE.NAME...... ETC.BP
 10. 9 RECO SUBR.ANC
  9.10 FILE.NAME...... ETC.BP
 10.10 RECO SUBR.CONV
  9.11 FILE.NAME...... ETC.BP
 10.11 RECO SUBR.INP
  9.12 FILE.NAME...... ETC.BP
 10.12 RECO SUBR.VAL
 31. 1 DATE.TIME...... 16 DEC 14 11:57
 32 AUTHORISER........ 24248_INPUTTER
 33 CO.CODE........... GB-001-0001         Model Bank R14
------------------------------------------------------------------------------
```

“BP” in ETC.BP allows us to save the source code as well - if it’s not there, directory name wouldn’t be accepted:

T24

```
Model Bank R14          DL.DEFINE INPUT

    UNIT.NAME......... TMNS000-QAZSW
------------------------------------------------------------------------------
  1. 1. 1 GB DESCRIPTN
  2. 1 GB SHORT.DESC..
  3 LANGUAGE/COUNTRY..
  4. 1 INDICES........
  5 OPERATION......... S
  6 SELECT.LIST.......
  7 TOP.LEVEL.TYPE....
  8 TOP.LEVEL.ITEM....
  9. 1 FILE.NAME...... MSG.IN              MISSING STANDARD.SELECTION - RECORD
 10. 1 RECO
 11. 1. 1 RECORD.DESC.
 12. 1 SAVED.FROM.....
 13. 1 SAVED.RELEASE..
 14. 1 SAVED.DATE.....
 15. 1 RESTORED.USER..
 16. 1 RESTD.COMPANY..
------------------------------------------------------------------------------
31 MAR 2015 10:28:56  USER (22 APR) VLADIMIR.K        [17315,IPAGE 1   >>>3>>>
ACTION
```

At this stage nothing is saved yet.  To save records open DL.DEFINE using V function and commit.  After that we can see the saved records in:

jsh

```
LIST ../F_DL_DATA/TMNS000-SAVE.20141219
```

Output

```
DL.D_TMNS000-SAVE.20141219
REC00001
REC00002
REC00003
REC00004
REC00005
REC00006
REC00007
REC00008
REC00009
REC00010
REC00011
REC00012
SS_EB.API
SS_ENQUIRY
SS_PGM.FILE
SS_VERSION
```

All files are text ones.  DL.D_TMNS000-SAVE.20141219 contains package information, REC...  are records, SS_...  are STANDARD.SELECTION records that hold corresponding application structure.

See REC0001 contents:

jsh

```
CT ../F_DL_DATA/TMNS000-SAVE.20141219 REC00001
```

Output

```
    REC00001
001
002 1
003 1
004
005
006
007
008
009
010
011
012
013 TRANSACTION.TYPE]DEBIT.VALUE.DATE]DEBIT.AMOUNT]DEBIT.CURRENCY]CREDIT.CURRE
    NCY
014 ]]]]
015 ]]]]
...
```

So - it’s a text copy of VERSION record FUNDS.TRANSFER,TRAIN - same would be created if we use a COPY from hashed file to UD file (i.e.  directory; note single quotes as comma would be considered a part of syntax otherwise):

jsh

```
COPY FROM F.VERSION TO &TEMP& 'FUNDS.TRANSFER,TRAIN'
CT &TEMP& FUNDS.TRANSFER,TRAIN
```

*(Then - to keep the temporary directory tidy - use command CLEAR-FILE &TEMP& in jsh.)*

To restore saved DL.DEFINE record at another environment, firstly use another “W” application DL.PARAMETER with command V SYSTEM:

T24 - target environment

```
Model Bank R14          Data Library Parameter File VERIFY

    ID.................. SYSTEM
------------------------------------------------------------------------------
  1. 1 REPLACE.CHARS..
  2. 1 REPLACE.WITH...
  3 OPERATION......... R
  4 FROM.FILE......... ../F.DL.DATA/DL.RESTORE
  5 OVERWRITING....... Y
  6. 1 RESTORE.UNITS.. TMNS000-SAVE.20141219  UNIT DOES NOT EXIST IN FROM.FILE
  7. 1 SAVE.UNITS.....
  8 TAPE.DEVICE.......
  9 RESERVED.9........
 10 RESERVED.8........
 11 RESERVED.7........
 12 RESERVED.6........
 13 RESERVED.5........
 14 RESERVED.4........
 15 RESERVED.3........
 16 RESERVED.2........
------------------------------------------------------------------------------
16 DEC 2014 12:14:57  USER (22 APR) INPUTTER          [32292,IPAGE 1   >>>2>>>
ACTION
```

*(Put the package to the folder specified in the field FROM.FILE to avoid the error message at the screen above.)*

After commit the record TMNS000-SAVE.20141219 will appear at DL.DEFINE. Use “Restore” operation with “V” function to restore the records.

T24

```
Model Bank R14 env.2    DL.DEFINE VERIFY

    UNIT.NAME......... TMNS000-SAVE.20141219
------------------------------------------------------------------------------
  1. 1. 1 GB DESCRIPTN First save ever
  2. 1 GB SHORT.DESC.. SAVE 1
  3 LANGUAGE/COUNTRY..
  4. 1 INDICES........
  5 OPERATION......... R
  6 SELECT.LIST.......
  ...
```

But that’s not all.  All application records (except “technical” ones like PGM.FILE, FILE.CONTROL etc) will be restored with “IHLD” status and have to be processed with input and authorisation; all source code files will be copied to ../DL_BP folder and have to be moved to corresponding “.BP” directory (which might or might not be the same as in the source environment) and then compiled.

*(When FILE.CONTROL record presents in the set, its target data files will be created automatically during the restore.)*

Another tip for DL.DEFINE. If you need to transfer many records from a single application, it’s not necessary to type all their @IDs into DL.DEFINE record.  For example, take INDUSTRY (one of several CUSTOMER-related applications):

jsh

```
COUNT FBNK.INDUSTRY
```

Output

```
 75 Records counted
```

SELECT comes to help here:

jsh

```
SELECT FBNK.INDUSTRY SAVING EVAL "'INDUSTRY>':@ID"
SAVE.LIST ALLIND
```

See the result that is in the format accepted by DL.DEFINE:

jsh

```
CT &SAVEDLISTS& ALLIND
```

Output

```
    ALLIND
001 INDUSTRY>4210
002 INDUSTRY>2460
003 INDUSTRY>1920
004 INDUSTRY>2600
005 INDUSTRY>2520
006 INDUSTRY>7000
...
```

*(We didn’t need the sort here but if you do - use SSELECT rather than SELECT or apply SORT.LIST to ALLIND in jsh.)*

Now open a new DL.DEFINE record and enter saved list name to the field SELECT.LIST. Result - 17 pages were populated:

T24

```
Model Bank R14          DL.DEFINE INPUT

    UNIT.NAME......... TMNS000-IND
------------------------------------------------------------------------------
  1. 1. 1 GB DESCRIPTN
  2. 1 GB SHORT.DESC..
  3 LANGUAGE/COUNTRY..
  4. 1 INDICES........
  5 OPERATION......... S
  6 SELECT.LIST.......
  7 TOP.LEVEL.TYPE.... _
  8 TOP.LEVEL.ITEM....
  9. 1 FILE.NAME...... INDUSTRY
 10. 1 RECO 4210
 11. 1. 1 RECORD.DESC.
  9. 2 FILE.NAME...... INDUSTRY
 10. 2 RECO 2460
 11. 2. 1 RECORD.DESC.
  9. 3 FILE.NAME...... INDUSTRY
 10. 3 RECO 1920
------------------------------------------------------------------------------
16 DEC 2014 12:35:23  USER (22 APR) INPUTTER          [12083,IPAGE 1   >>17>>>
ACTION
```

The last multi-value association has the number 75 - the same as number of records in INDUSTRY LIVE file:

T24

```
Model Bank R14          DL.DEFINE INPUT

    UNIT.NAME......... TMNS000-IND
--------------------------------------------------------------------------------
  9.73 FILE.NAME...... INDUSTRY
 10.73 RECO 2300
 11.73. 1 RECORD.DESC.
  9.74 FILE.NAME...... INDUSTRY
 10.74 RECO 3300
 11.74. 1 RECORD.DESC.
  9.75 FILE.NAME...... INDUSTRY
 10.75 RECO 3400
 11.75. 1 RECORD.DESC.
 12. 1 SAVED.FROM.....
 13. 1 SAVED.RELEASE..
 14. 1 SAVED.DATE.....
 15. 1 RESTORED.USER..
 16. 1 RESTD.COMPANY..
 17. 1 RESTD.RELEASE..
 18. 1 RESTD.DATE.....
--------------------------------------------------------------------------------
16 DEC 2014 12:37:01  USER (22 APR) INPUTTER          [12083,IPAGE 15  >>17>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

If we want to add another application - it’s easy.  Save @IDs of another CUSTOMER-related application TARGET (12 more records) and concatenate 2 lists:

jsh

```
SELECT FBNK.TARGET SAVING EVAL "'TARGET>':@ID"
SAVE.LIST ALLTG
OR-LISTS ALLIND ALLTG
SAVE.LIST ALL
```

Result

```
87 record(s) saved to list 'ALL'
```

*(There is also utility AND-LISTS that puts to the result only records that present in both lists - of course giving zero result in this case.)*

Sort this list and see the end of it using (N option - no stop on pages:

jsh

```
SORT.LIST ALL
CT &SAVEDLISTS& ALL (N
```

Output

```
...
072 INDUSTRY>8190
073 INDUSTRY>8200
074 INDUSTRY>9000
075 INDUSTRY>9999
076 TARGET>1
077 TARGET>10
078 TARGET>2
079 TARGET>20
080 TARGET>3
081 TARGET>30
082 TARGET>4
083 TARGET>50
084 TARGET>6
085 TARGET>7
086 TARGET>999
087 TARGET>9999
```

*(Note that the sort is a text one - left-to-right - even when @IDs are numeric.)*

Important note about DL.DEFINE: though technically records of all T24 applications can be transferred via DL.DEFINE, there are ones that should not be, e.g. ACCOUNT whose records contain financial data like WORKING.BALANCE. If they are transferred to another environment as they are, that data won't match the other financial data on target environment.

## 1.55 Local fields

There is a possibility for a local developer to add new fields to T24 applications, even core ones. Such fields are called local and they all are stored in the field that many (but not all) T24 applications have for this purpose. This field is called LOCAL.REF.

Local fields are delimited by @VM so they can be multi-valued like "regular" ones but can't also be subvalued or "language" ones.

Local fields are set up in 2 applications. Field format is defined in LOCAL.TABLE:

T24

```
Model Bank R14          LOCAL.TABLE SEE

    TABLE.NO.......... CU.EFF.DATE
--------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. CU.EFF.DATE
  2. 1 GB SHORT.NAME.. CU.EFF.DATE
  3 MAXIMUM.CHAR...... 11
  5 CHAR.TYPE......... D
 32 CURR.NO........... 1
 33. 1 INPUTTER....... 1_201012m
 34. 1 DATE.TIME...... 28 MAR 14 13:14
 35 AUTHORISER........ 98220_INPUTTER_OFS_MB.OFS.AUTH
 36 CO.CODE........... GB-001-0001         Model Bank R14
 37 DEPT.CODE......... 1                   Implementation
```

The field is attached to application in LOCAL.REF.TABLE, see where:

T24

```
LRT L L
```

Selection criterion

```
1 EQ CU.EFF.DATE
```

CUSTOMER is the only result.  See it:

T24

```
Model Bank R14          LOCAL.REF.TABLE SEE

    APPLICATION....... CUSTOMER            CUSTOMER
--------------------------------------------------------------------------------
  1. 1 LOCAL.TABLE.NO. CUSTOMER.SEGMENT    SEGMENT
  1. 2 LOCAL.TABLE.NO. CU.EFF.DATE         CU.EFF.DATE
  1. 3 LOCAL.TABLE.NO. WELCOME.PACK        WELCOME.PACK
  4 CURR.NO........... 2
  5. 1 INPUTTER....... 1094_INPUTTER__OFS_BROWSERTC
  6. 1 DATE.TIME...... 31 MAY 14 11:03
  7 AUTHORISER........ 1094_INPUTTER_OFS_BROWSERTC
  8 CO.CODE........... GB-001-0001         Model Bank R14
  9 DEPT.CODE......... 1                   Implementation









--------------------------------------------------------------------------------
16 DEC 2014 08:51:36  USER (22 APR) INPUTTER          [19612,IPAGE 1
ACTION
AWAITING PAGE INSTRUCTIONS
```

See these 3 local fields in CUSTOMER record:

T24

```
Model Bank R14          CUSTOMER INPUT

    CUSTOMER.CODE..... 100226
--------------------------------------------------------------------------------
...
179. 1 SEGMENT........                                                      <=====
179. 2 CU.EFF.DATE....                                                      <=====
179. 3 WELCOME.PACK...                                                      <=====
180. 1 OVERRIDE....... KYCDOCS/CUS*410 FROM 100226 NOT RECEIVED
180. 2 OVERRIDE....... PWMDOCS/CUS*410 FROM 100226 NOT RECEIVED
180. 3 OVERRIDE....... INTRO/CUS*410 FROM 100226 NOT RECEIVED
181 RECORD.STATUS..... INAU                INPUT Unauthorised
...
```

To see how to address local fields in VERSION hook routines we create a VERSION for CUSTOMER and CHECK.REC.RTN for it.  We'll raise the error if the local field CU.EFF.DATE is empty so user won't be able to edit such record.

Routine source code:

jBC

```
SUBROUTINE SUBR.CRR

$INSERT I_COMMON
$INSERT I_EQUATE

* APPLICATION is a global variable.
* Its usage makes the routine usable for all applications
   CALL GET.LOC.REF(APPLICATION, 'CU.EFF.DATE', posn)

* We don't need the INSERT file to get LOCAL.REF field position
   cu_eff_date = R.NEW(LOCAL.REF.FIELD)<1,posn>

* note <1,posn> - local fields are @VM-delimited
   IF cu_eff_date EQ '' THEN E = 'EB-BLANK.FLD.INVALID'

   RETURN
END
```

Note that in different places we use different technique to raise an error:

- Validation routine:  set ETEXT.
- Input routine:  set AF and ETEXT, call core subroutine STORE.END.ERROR.
- Check-record routine above:  set E.

We'll also need to create EB.API record and to attach the routine to a VERSION.

Run that VERSION with a record that doesn't have the local field CU.EFF.DATE populated:

T24

```
-------------------------------------------------------------------------------
16 DEC 2014 13:53:12  USER (22 APR) INPUTTER          [23019,IN]
ACTION 100226                                             Blank Field Invalid
AWAITING ID
```

The error message text is taken from EB.ERROR record with @ID that we supplied for E global variable:

T24

```
Model Bank R14          GLOBUS Error Message Table SEE

    EB.ERROR.ID....... EB-BLANK.FLD.INVALID
------------------------------------------------------------------------------
  1. 1. 1 L Blank Field Invalid
  1. 1. 2 L Champ vide non valide
 11 NUMERIC.ID........ E-103317
 21 CURR.NO........... 3
 22. 1 INPUTTER....... 40608_INPUTTER__OFS_MB.LANG
 23. 1 DATE.TIME...... 02 JUN 14 16:46
 24 AUTHORISER........ 40608_INPUTTER_OFS_MB.LANG
 25 CO.CODE........... GB-001-0001         Model Bank R14
 26 DEPT.CODE......... 1                   Implementation









------------------------------------------------------------------------------
16 DEC 2014 13:55:00  USER (22 APR) INPUTTER          [23019,IPAGE 1
ACTION
AWAITING PAGE INSTRUCTIONS
```

Error message translation is also here; if we log in using user profile with language set as 2...

T24

```
Model Bank R14          USER PROFILE, INPUT

    USER.ID........... VLADIMIR.K1
------------------------------------------------------------------------------
  1 USER.NAME......... VLADIMIR K
  2 SIGN.ON.NAME...... VLADIMIRK1
  3 CLASSIFICATION.... INT
  4 LANGUAGE.......... 2                   French
  ...
```

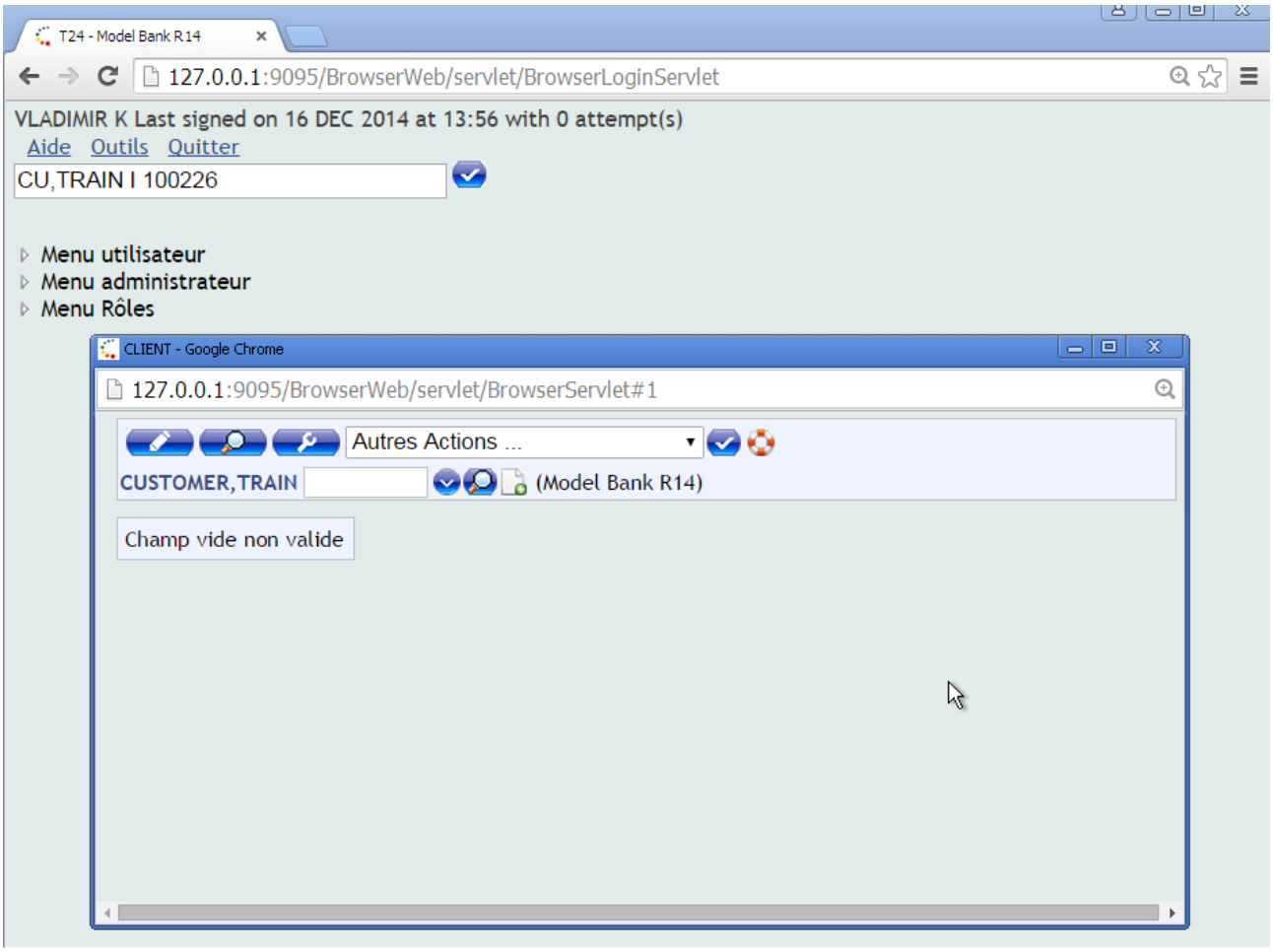


**Figure 39: Error message in French.**

*(Using S function we are able to see the record. It means that CHECK.REC.RTN isn't triggered in this case.)*

We already saw an example of fields like the first one of EB.ERROR that are called "language fields". We can also see them, e.g., in CUSTOMER:

T24

```
Model Bank R14          CUSTOMER INPUT

    CUSTOMER.CODE..... 111615
------------------------------------------------------------------------------
  1 MNEMONIC.......... ABCE
  2. 1 GB SHORT.NAME.. ABC EUROPE
  3. 1 GB NAME.1...... ABC EUROPE
  4. 1 GB NAME.2...... ABC EUROPE
  5. 1 GB STREET...... FRANKFURTER STR.107
  5. 2 FR STREET...... D-64807 DIEBURG
  6. 1. 1 GB ADDRESS..
  7. 1 GB TOWN.COUNTRY GERMANY
  8. 1 GB POST.CODE...
  9. 1 GB COUNTRY.....
  ...
```

...and it’s defined in the corresponding SS record:

T24

```
 Model Bank R14          STANDARD SELECTION FIELDS SEE

     FILE.NAME......... CUSTOMER
 ------------------------------------------------------------------------------
  11. 4 SYS.LANG.FIELD. N
  12. 4 SYS.GENERATED.. Y
   1. 5 SYS.FIELD.NAME. SHORT.NAME
   2. 5 SYS.TYPE....... D
   3. 5. 1 SYS.FIELD.NO 2
   4. 5. 1 SYS.VAL.PROG IN2SWI
   6. 5 SYS.DISPLAY.FMT 35L
   7. 5 SYS.ALT.INDEX.. N
  10. 5 SYS.SINGLE.MULT M
  11. 5 SYS.LANG.FIELD. Y                                        <=====
  12. 5 SYS.GENERATED.. Y
   1. 6 SYS.FIELD.NAME. NAME.1
   2. 6 SYS.TYPE....... D
   3. 6. 1 SYS.FIELD.NO 3
   4. 6. 1 SYS.VAL.PROG IN2SWI
   6. 6 SYS.DISPLAY.FMT 35L
 ------------------------------------------------------------------------------
 06 MAR 2015 12:57:09  USER (22 APR) VLADIMIR.K        [24033,IPAGE 3   >>116>>
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

Back to EB.ERROR. There is a record in EB.ERROR which produced the message that we already saw:

jsh

```
LIST F.EB.ERROR 'EB-APP.UNDEFINED.SUBR' ERROR.MSG
```

Output

```
@ID....... EB-APP.UNDEFINED.SUBR
ERROR.MSG. APPL NOT DEFINED FOR THIS SUBR   APPL. NON DEFINIE POUR CETTE
      SOUS-ROUTINE
```

Finally, we need futher tests to prove that our routine works correctly.  A minimum test would be to find a record where that local field is populated and open it with our VERSION using function I.

T24 “AWAITING APPLICATION” prompt

```
CU,TRAIN L L
```

Selection criterion

```
CU.EFF.DATE NE ''
```

Results

```
Model Bank R14


------------------------------------------------------------------------------
1  100400 BOSCHR     Robert Bosch Jr                PWM Portfolio Advis|  G|  G|  I
2  100297 BRANSONR   Robert Branson                 PWM Relationship Ma|  G|  G|  I
3  100256 DELTA      Delta Caroline                 Corporate Loan Supe|  U|  U|  I
4  100407 HSCHWARZ   Herman Schwarz                 PWM Relationship Ma|  G|  E|  *
5  100271 PEUGEOT    Peugeot Family                 PWM Relationship Ma|  F|  F|  I
6  129025 RBOSSR     Robert Bosch Sr                PWM Portfolio Advis|  G|  G|  I
7  100273 SAINSBURYD David Sainsbury                PWM Portfolio Advis|  G|  G|  I









------------------------------------------------------------------------------
16 DEC 2014 14:11:34  USER (22 APR) INPUTTER          [13865,IPAGE   1 >>>1>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

Edit the first record, go to the record end (F4):

T24

```
Model Bank R14          CUSTOMER,TRAIN INPUT

    CUSTOMER.CODE..... 100400
------------------------------------------------------------------------------
177 RESERVED.02.......
178 RESERVED.01.......
179. 1 SEGMENT........
179. 2 CU.EFF.DATE.... 01 JUL 2005
179. 3 WELCOME.PACK...
180. 1 OVERRIDE.......
181 RECORD.STATUS.....
182 CURR.NO........... 1
183. 1 INPUTTER....... 6656_OFFICER__OFS_SEAT
184. 1 DATE.TIME...... 08 APR 14 17:10
185 AUTHORISER........ 6656_OFFICER_OFS_SEAT
186 CO.CODE........... GB-001-0001         Model Bank R14
187 DEPT.CODE......... 1                   Implementation
188 AUDITOR.CODE......
189 AUDIT.DATE.TIME...

------------------------------------------------------------------------------
16 DEC 2014 14:12:40  USER (22 APR) INPUTTER          [13865,IPAGE 12
ACTION
AWAITING PAGE INSTRUCTIONS
```

*(More complex routines are to be debugged of course.)*

Still the routine isn’t good enough to work in production environment.  Let’s assume a typo in local field name:

jBC

```
   CALL GET.LOC.REF(APPLICATION, 'CU.EFFF.DATE', posn)
                                  ^^^^
```

Try it on the record 100226 and it now lets us in.  Add debugger statement at the very beginning to see why.  When we stop, debugger command W shows the source code (if it doesn’t - use command P to indicate source code location, e.g.  P ETC.BP and then W again):

Debugger

```
0002     $INSERT I_COMMON
0003     $INSERT I_EQUATE
0004 * APPLICATION is a global variable.
0005 * Its usage makes the routine usable for all applications
0006 DEBUG
0007     CALL GET.LOC.REF(APPLICATION, 'CU.EFFF.DATE', posn)
0008 * We don't need the INSERT file to get LOCAL.REF field position
0009     cu_eff_date = R.NEW(LOCAL.REF.FIELD)<1,posn>
0010 * note <1,posn> - local fields are @VM-delimited
```

To move one step forward enter command S to stay on the same level; lowercase s will bring us inside GET.LOC.REF whose source code we won’t be able to see anyway (full list of debugger commands - type ?).

Now we are positioned on CALL GET.LOC.REF line, see what’s there in variable *posn*:

Debugger

```
0007     CALL GET.LOC.REF(APPLICATION, 'CU.EFFF.DATE', posn)
jBASE debugger->V posn
  posn                      : (UNASSIGNED)
```

Another step forward and see it again:

Debugger

```
0009     cu_eff_date = R.NEW(LOCAL.REF.FIELD)<1,posn>
jBASE debugger->V posn
  posn                      : (UNASSIGNED)
```

So GET.LOC.REF doesn’t assign any value to variable *posn* in case of an error.

Maybe it sets up one of global variables that are normally used for error messages or codes?

Debugger

```
jBASE debugger->V ETEXT
COMMON variables
  ETEXT                    :
jBASE debugger->V E
COMMON variables
  E                        :
```

Nope...  but why we were allowed to open the record?  Take another step, then see what's in variable *cu_eff_date*:

Debugger

```
jBASE debugger->V cu_eff_date
  cu_eff_date              : ]
```

What was returned from the expression R.NEW(LOCAL.REF.FIELD)<1,posn> is full LOCAL.REF field content:

Debugger

```
jBASE debugger->V LOCAL.REF.FIELD
COMMON variables
  LOCAL.REF.FIELD          : 179
jBASE debugger->V R.NEW(179)
COMMON variables
  R.NEW(179)               : ]
```

This content is then checked for being empty and - since it's not - we are allowed to edit the record.

As a one-time correction (and double-checker) we can amend the value of a variable in debugger - in this case to remove the value altogether...

Debugger

```
jBASE debugger->V -m cu_eff_date
  cu_eff_date              : ] = \0
jBASE debugger->V cu_eff_date
  cu_eff_date              :
```

...and then enter debugger command C to continue execution:

T24

```
jBASE de100226                                          Blank Field Invalid
 AWAITING ID
```

Another question - why we didn’t get any error message about unassigned variable? See some settings:

jsh

```
echo %JBASE_ERRMSG_NON_NUMERIC%
echo %JBASE_ERRMSG_ZERO_USED%
echo %JBASE_ERRMSG_DIVIDE_BY_ZERO%
```

Output

```
19
35
19
```

T24 by default is set up to suppress such error messages even in Classic mode (in Browser they wouldn’t be seen in any case).  To correct this situation we need to edit the remote.cmd:

jsh

```
JED . remote.cmd
```

*(Or - if remote.cmd calls another file like mbremote.cmd - edit the latter.)*

Search these 3 items - without percent signs of course; if they exist set them to 0, if not - add them:

JED - remote.cmd

```
0080 set JBASE_ERRMSG_NON_NUMERIC=0
0081 set JBASE_ERRMSG_ZERO_USED=0
0082 set JBASE_ERRMSG_DIVIDE_BY_ZERO=0
```

Restart the telnet/ssh session to renew the environment and see the difference:

Debugger

```
 ------------------------------------------------------------------------
 ---DEBUG statement seen
Source changed to D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\ETC.BP\SUBR.CRR
0006 DEBUG
jBASE debugger->S
0007     CALL GET.LOC.REF(APPLICATION, 'CU.EFFF.DATE', posn)
jBASE debugger->S
0009     cu_eff_date = R.NEW(LOCAL.REF.FIELD)<1,posn>
jBASE debugger->S
Non-numeric value -- ZERO USED ,                                    <==== here
Variable 'posn' , Line     9 , Source SUBR.CRR
```

The final solution then will be to assign a “erroneous” value to variable *posn* and if after call to GET.LOC.REF it’s still the same - raise another error:

jBC

```
SUBROUTINE SUBR.CRR

$INSERT I_COMMON
$INSERT I_EQUATE

   loc_fld_name = 'CU.EFFF.DATE'
   posn = -1
   CALL GET.LOC.REF(APPLICATION, loc_fld_name, posn)
   IF posn EQ -1 THEN
      E = 'DC-MISS.LOCAL.REF.TABLE' :FM: loc_fld_name
      RETURN
   END

   cu_eff_date = R.NEW(LOCAL.REF.FIELD)<1,posn>

   IF cu_eff_date EQ '' THEN E = 'EB-BLANK.FLD.INVALID'

   RETURN
END
```

Try it:

T24

```
 ------------------------------------------------------------------------------
 16 DEC 2014 15:26:14  USER (22 APR) INPUTTER          [18488,IN]
 ACTION 100226                    CU.EFFF.DATE  is missing from LOCAL.REF.TABLE
 AWAITING ID
```

Now correct the typo and test again.

*(This logic would be handy if the local field is a new one and wasn’t yet created in the current environment.)*

To add your own local field to an application open the corresponding LOCAL.REF.TABLE record, e.g.  CUSTOMER. Go to field 2.3, add new multi-value association.

To choose from a list in Classic interface input the value !  to the field 1.4:

T24

```
Model Bank R14          LOCAL.REF.TABLE, INPUT

    APPLICATION....... CUSTOMER            CUSTOMER
--------------------------------------------------------------------------------
Model Bank R14          LOCAL.TABLE


--------------------------------------------------------------------------------









--------------------------------------------------------------------------------
16 DEC 2014 16:11:05  USER (22 APR) INPUTTER          [27220,
ACTION _
AWAITING FUNCTION
```

We’re now in another T24 application - LOCAL.TABLE. Type L to see the records (read - choices) list:

T24

```
Model Bank R14          LOCAL.REF.TABLE, INPUT

    APPLICATION....... CUSTOMER            CUSTOMER
--------------------------------------------------------------------------------
Model Bank R14           LOCAL.TABLE - Default List

   ID                                    SHORT.NAME       MAXIMUM.CHAR   FUNCT.
--------------------------------------------------------------------------------
 1  151                                   IBLC CODE-DR     3
 2  152                                   IBLC CODE-CR     3
 3  153                                   IBLC CODE-CH     3
 4  170                                   IBLC COUNTRY     3
 5  171                                   IBLC COUNTRY-DR  3
 6  172                                   IBLC COUNTRY-CR  3
 7  173                                   IBLC COUNTRY-CH  3
 8  AA.DOC                                TXNID            65
 9  APP.FORM.ID                           APP.FORM.ID      25
10  ARRANGEMENT.ID                        ARRANGEMENT      16
11  BC.SORT.CODE                          BC.SORT.CODE     20
12  BEN.ALT.KEY                           BEN.ALT.KEY      20
--------------------------------------------------------------------------------
16 DEC 2014 16:12:25  USER (22 APR) INPUTTER          [27220, PAGE   1 >>>3>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

You can navigate through pages using F3/F2. To choose a record, e.g. record EXTERNAL.REFERENCE.NO which is #1 at page 3, go to that page (F3 F3 or - directly - P3, then type 1 Enter C Enter F5. We're back:

T24

```
Model Bank R14          LOCAL.REF.TABLE, INPUT

    APPLICATION....... CUSTOMER            CUSTOMER
------------------------------------------------------------------------------
  1. 1 LOCAL.TABLE.NO. CUSTOMER.SEGMENT    SEGMENT
  2. 1 SUB.ASSOC.CODE.
  1. 2 LOCAL.TABLE.NO. CU.EFF.DATE         CU.EFF.DATE
  2. 2 SUB.ASSOC.CODE.
  1. 3 LOCAL.TABLE.NO. WELCOME.PACK        WELCOME.PACK
  2. 3 SUB.ASSOC.CODE.
  1. 4 LOCAL.TABLE.NO. EXTERNAL.REFERENCE.NO EXT.REF    <===== chosen value here
  2. 4 SUB.ASSOC.CODE. _                                  - together with enrichment
  3 RECORD.STATUS.....
  4 CURR.NO........... 2
  5. 1 INPUTTER....... 1094_INPUTTER__OFS_BROWSERTC
  6. 1 DATE.TIME...... 31 MAY 14 11:03
  7 AUTHORISER........ 1094_INPUTTER_OFS_BROWSERTC
  8 CO.CODE........... GB-001-0001         Model Bank R14
  9 DEPT.CODE......... 1                   Implementation
 10 AUDITOR.CODE......
------------------------------------------------------------------------------
16 DEC 2014 16:15:12  USER (22 APR) INPUTTER          [27220, PAGE 1   >>>2>>>
ACTION
```

Field 2.4 - SUB.ASSOC.CODE - is to be populated if we want this local field to be multi-valued. Use "XX." if the field is by itself and "XX<" ... "XX-" ... "XX>" for several fields that are part of a multi-value association.

*(After commit and authorisation the local field can't be removed or changed to another one - only a low-level procedure can assist in that. Field SUB.ASSOC.CODE can be amended though. So it's crucial to have the order of local fields to be the same in all T24 environments - otherwise there would be a mess when records are transferred from one environment to another.)*

## 1.56 Calling other routines

The main thing to remember is to call with exactly the same number of parameters that is declared in called subroutine. We already called T24 API subroutine REM with no parameters. In that case the data exchange was done via global (COMMON) area declared in I_COMMON.

IN2D routine was called with 2 parameters. If we try to call a subroutine with different number of parameters...

See:

jBC

```
SUBROUTINE SUBR.ANC

$INSERT I_COMMON
$INSERT I_EQUATE

   DEBUG
   COMI = R.NEW(AF)
   CALL IN2A('54.3', 'A', 'test')

   RETURN
END
```

Launch VERSION again to the point where we are in debugger.  Type S to proceed by a single step (or C to continue execution):

Debugger

```
jBASE debugger->C
 ** Error [ SUBROUTINE_PARM_ERROR ] **
'SUBROUTINE IN2A' called with incorrect arguments , Line     1 , Source SUBR.ANC
Trap from an error message, error message name = SUBROUTINE_PARM_ERROR
jBASE debugger->
```

That’s why additional option for IN2D to handle years before 1950 was done using dynamic array so number of parameters remained the same:

jBC

```
CALL IN2D(11, 'D' :@FM: 1000)
```

Parameter that is passed to a subroutine can be changed by it unless the parameter is passed by value rather than by reference.  Example:

jBC

```
SUBROUTINE SUBR.0
* T24 mainline routine
$INSERT I_COMMON
$INSERT I_EQUATE
   cur_date = TODAY
   TEXT = cur_date  ;  CALL REM

* see which day is 42 days away
   CALL CDT('', cur_date, '42C')
   TEXT = cur_date  ;  CALL REM

* another call but cur_date won't change
   CALL CDT('', (cur_date), '42C')
*               ^        ^---------------------------- call by value
   TEXT = cur_date  ;  CALL REM
   RETURN
END
```

Output

```
 CONTINUE (Y)                                                    20140422
...
 CONTINUE (Y)                                                    20140603
...
 CONTINUE (Y)                                                    20140603
```

*(TODAY should never be changed in local developments so don’t use it in subroutine parameters to be on the safe side.)*

Subroutine can be called by name or by reference:

jBC

```
...
   CALL CDT('', cur_date, '42C')
...
* or:
...
   rtn_name = 'CDT'
   CALL @rtn_name('', cur_date, '42C')
...
```

By reference are usually called T24 hook routines that are defined in VERSION, ENQUIRY records etc.  If we try to call a nonexistent subroutine, the error occurres:

jBC program

```
the_subr = 'QWERTY'
CALL @the_subr
```

Output

```
 ** Error [ SUBROUTINE_CALL_FAIL ] **
Unable to perform CALL to subroutine QWERTY , Line     2 , Source PROG.0
Press C to continue or Q to quit
Trap from an error message, error message name = SUBROUTINE_CALL_FAIL
Source changed to D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\ETC.BP\PROG.0
0002 CALL @the_subr
jBASE debugger->
```

*(If active jbase agent handles the request that involves a call to a subroutine that was created and compiled after its start, it will fail with the same error.)*

To check whether a subroutine exists, use JBASESubroutineExist() function:

jBC program

```
the_subr = 'QWERTY'
GOSUB CHECK.IT

the_subr = 'CDD'
GOSUB CHECK.IT

STOP

CHECK.IT:
   result = CALLC JBASESubroutineExist(the_subr, subr_info)
   IF result EQ 1 THEN CRT the_subr, 'exists' ELSE CRT 'No such subroutine', the_subr
   RETURN
END
```

*(When calling a subroutine, especially a core one, it's a good idea to initialize all parameters that are passed to it.  We saw it earlier in GET.LOC.REF call example.)*

## 1.57 I-descriptors

When we take a closer look at STANDARD.SELECTION records we can see there not only data fields (SYS.TYPE=D) but ones with SYS.TYPE=I:

T24

```
Model Bank R14          STANDARD SELECTION FIELDS SEE

    FILE.NAME......... CONTEXT.ENQUIRY
--------------------------------------------------------------------------------
 11. 2 SYS.LANG.FIELD. N
 12. 2 SYS.GENERATED.. Y
  1. 3 SYS.FIELD.NAME. PGM.VERSION
  2. 3 SYS.TYPE....... I                                          <=====
  3. 3. 1 SYS.FIELD.NO @ID["-",1,1]
  6. 3 SYS.DISPLAY.FMT 35L
 10. 3 SYS.SINGLE.MULT S
  1. 4 SYS.FIELD.NAME. FIELD
  2. 4 SYS.TYPE....... I                                          <=====
  3. 4. 1 SYS.FIELD.NO @ID["-",2,1]
  6. 4 SYS.DISPLAY.FMT 35L
 10. 4 SYS.SINGLE.MULT S
  1. 5 SYS.FIELD.NAME. DESCRIPTION
  2. 5 SYS.TYPE....... D
  3. 5. 1 SYS.FIELD.NO 1
  4. 5. 1 SYS.VAL.PROG IN2A
--------------------------------------------------------------------------------
08 OCT 2014 03:36:34  USER (22 APR) INPUTTER          [9637,INPAGE 2   >>22>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

They are so-called "I-descriptors" and it's standard functionality in multi-value databases.  We can consider them as calculated fields.

I-descriptors can be used in jQL queries, in SELECTs for enquiry FIXED.SELECTION, they can also be diplayed in enquiries.  Example of using the I-descriptor PGM.VERSION that was shown at the previous screen:

jsh

```
LIST F.CONTEXT.ENQUIRY LIKE "...'-'..." PGM.VERSION
```

Output

```
@ID......... FUNDS.TRANSFER,IT.LCYVOST.FTHP-DEBIT.ACCT.NO
PGM.VERSION. FUNDS.TRANSFER,IT.LCYVOST.FTHP

@ID......... SL.LOANS-DELIVERY.REF
PGM.VERSION. SL.LOANS

@ID......... AM.COMPARE.DETAIL-GRID.CODE
PGM.VERSION. AM.COMPARE.DETAIL

@ID......... DIARY-SECURITY.NO
PGM.VERSION. DIARY

@ID......... PAYMENT.STOP-ANSWERS
PGM.VERSION. PAYMENT.STOP
...
```

Output above is the same as for EVAL "FIELD(@ID,’-’,1)".

If we look at the screen after T24 command CONTEXT.ENQUIRY L L we can see this I-descriptor in fields list:

T24

```
...
 35. AUDIT.DATE.TIME       X.. PGM.VERSION          XA. FIELD
...
```

So typing selection X LK ...STOP brings us the following result:

T24

```
   ID                                                                  FUNCT.
 ------------------------------------------------------------------------------
 1  PAYMENT.STOP-ANSWERS
...
```

Other examples.

Just a synonim (for compatibility sake, possibly for accounting consolidation key creation):

T24 SS record - ACCOUNT

```
  1.15 SYS.FIELD.NAME. PRODCCY
  2.15 SYS.TYPE....... I
  3.15. 1 SYS.FIELD.NO CURRENCY
  6.15 SYS.DISPLAY.FMT 3L
 10.15 SYS.SINGLE.MULT S
```

Part of @ID is extracted:

T24 SS record - ACCOUNT

```
  1102 SYS.FIELD.NAME. TELLER.ID
  2102 SYS.TYPE....... I
  3102. 1 SYS.FIELD.NO @ID[9,4]
  6102 SYS.DISPLAY.FMT 4R
 10102 SYS.SINGLE.MULT S
```

*(As you can see, when the MV number exceeds 99, the dot is sacrificed to keep the screen format.  Still you can type something like 3.102.1 to get to the field in edit mode.)*

Fields concatenation:

T24 SS record - CUSTOMER

```
  1.42 SYS.FIELD.NAME. NAME.ADDRESS
  2.42 SYS.TYPE....... I
  3.42. 1 SYS.FIELD.NO SHORT.NAME:' ':NAME.1:' ':NAME.2:' ':STREET:' '
  3.42. 2 SYS.FIELD.NO :TOWN.COUNTRY
  6.42 SYS.DISPLAY.FMT 50L
 10.42 SYS.SINGLE.MULT S
```

*(Field SYS.FIELD.NO has subvalues in case the expression is longer than its length.)*

If-then-else statement:

T24 SS record - CUSTOMER

```
  1.41 SYS.FIELD.NAME. CHECK.LIAB
  2.41 SYS.TYPE....... I
  3.41. 1 SYS.FIELD.NO CUSTOMER.LIABILITY ; IF @1 NE "" AND @1 NE @ID
  3.41. 2 SYS.FIELD.NO  THEN "" ELSE @ID
  6.41 SYS.DISPLAY.FMT 10L
 10.41 SYS.SINGLE.MULT S
```

*(@1 here is the result of previous evaluation, i.e.  CUSTOMER.LIABILITY.)*

Local fields are also I-descriptors:

T24

```
Model Bank R14          STANDARD SELECTION FIELDS SEE

    FILE.NAME......... CUSTOMER
--------------------------------------------------------------------------------
 24. 1 USR.SINGLE.MULT S
 25. 1 USR.LANG.FIELD. N
 27. 1. 1 USR.REL.FILE EB.CUSTOMER.SEGMENT Customer Segmentation
 15. 2 USR.FIELD.NAME. CU.EFF.DATE
 16. 2 USR.TYPE....... I
 17. 2. 1 USR.FIELD.NO LOCAL.REF<1,2>
 18. 2. 1 USR.VAL.PROG IN2D
 20. 2 USR.DISPLAY.FMT 11R
 24. 2 USR.SINGLE.MULT S
 25. 2 USR.LANG.FIELD. N
 15. 3 USR.FIELD.NAME. WELCOME.PACK
 16. 3 USR.TYPE....... I
 17. 3. 1 USR.FIELD.NO LOCAL.REF<1,3>
 18. 3. 1 USR.VAL.PROG IN2A
 20. 3 USR.DISPLAY.FMT 3L
 24. 3 USR.SINGLE.MULT S
--------------------------------------------------------------------------------
16 DEC 2014 17:02:00  USER (22 APR) INPUTTER          [19745,IPAGE 115 >>116>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

*(For example, second local field CU.EFF.DATE is defined as LOCAL.REF<1,2> - the second value of field LOCAL.REF.)*

*(SYS.FIELD.NO is replaced with USR.FIELD.NO in this part of SS record, other fields are renamed as well; that’s where local I-descriptors are to be created.)*

## 1.58 Usage of routines in I-descriptors

When the logic of calculations is more complex than in the examples above, routine can be attached to solve this task.  In the following example the @IDs of all ADI records (shortcut for ACCOUNT.DEBIT.INT) are retrieved:

T24

```
Model Bank R14          STANDARD SELECTION FIELDS SEE

    FILE.NAME......... ACCOUNT
--------------------------------------------------------------------------------
  1. 5 SYS.FIELD.NAME. ADI.ID
  2. 5 SYS.TYPE....... I
  3. 5. 1 SYS.FIELD.NO SUBR("AC.GET.CRDR.ID","DR",ACCT.ID)
  6. 5 SYS.DISPLAY.FMT 25L
 10. 5 SYS.SINGLE.MULT M
```

So:  AC L L, selection:  ADI.ID NE ”

Result

```
     No records were found that matched the selection criteria

     SSELECT FBNK.ACCOUNT WITH   ADI.ID NE "" AND
     CO.CODE = "GB0010001"  BY CATEGORY BY CURRENCY
```

No such accounts found in the current company.  If we were logged in to T24 in the current session, we can also use this field in jQL query:

jsh

```
LIST FBNK.ACCOUNT ADI.ID WITH ADI.ID NE ''
```

Output

```
@ID.............    ADI.ID..................

              66044    66044-20140324
              66621    66621-20140324
              12165    12165-20140324
              66079    66079-20140324
              66068    66068-20140324
              66052    66052-20140324

 6 Records Listed
```

If a “fresh” session output will be:

Output

```
    0** FATAL ERROR IN (OPF) **
MISSING FILE - F0.ACCOUNT , MNEMONIC = 0 , IN.PARAMETER = F.ACCOUNT , COMPANY = 0 ,
CALL.ROUTINE = OPF,AC.GET.CRDR.ID,AC.GET.CRDR.ID
```

Why we need T24 login first?  T24 global areas are not cleared upon log off and they were used in the subroutine.

What is special in these ADI records?  @ID contains the effective date for the debit interest and we can’t take the latest one because it might contain a future date:

jsh

```
LIST ONLY FBNK.ACCOUNT.DEBIT.INT LIKE "'66044-'..."
```

Output

```
66044-20140324
```

There’s only one record which is effective from 24/03/2014.  We can add another one effective from TODAY that is 22/04/2014.

To default the effective date to TODAY just type account @ID and populate some data:

T24

```
Model Bank R14          Account Debit Interest, INPUT

    ACCOUNT.NO.DATE... 66044 - 22 APR 2014 Randall Hawley
------------------------------------------------------------------------------
  1 CHARGE.KEY........ 1                   Current Account (Personal) Regime
  2 INTEREST.DAY.BASIS B                   366/360
  3 TAX.KEY...........
  4 DR.BALANCE.TYPE... AVERAGE
  5 DR.CALCUL.TYPE....
  6. 1 DR.BASIC.RATE..
  7. 1 DR.INT.RATE.... 15.00
  8. 1 DR.MARGIN.OPER.
  9. 1 DR.MIN.RATE.... 10.00
 10. 1 DR.MARGIN.RATE.
 11. 1 DR.LIMIT.AMT...
 12. 1 DR.MAX.RATE....
 13 DR2.BALANCE.TYPE..
 14 DR2.CALCUL.TYPE...
 15. 1 DR2.BASIC.RATE.
 16. 1 DR2.INT.RATE...
------------------------------------------------------------------------------
16 DEC 2014 17:41:22  USER (22 APR) INPUTTER          [8572,I PAGE 1   >>>3>>>
ACTION
```

We can also add some future record:

T24

```
Model Bank R14          Account Debit Interest, INPUT

    ACCOUNT.NO.DATE... 66044 - 01 JAN 2015 Randall Hawley
------------------------------------------------------------------------------
  1 CHARGE.KEY........ 1                   Current Account (Personal) Regime
  2 INTEREST.DAY.BASIS B                   366/360
  3 TAX.KEY...........
  4 DR.BALANCE.TYPE... AVERAGE
  5 DR.CALCUL.TYPE....
  6. 1 DR.BASIC.RATE..
  7. 1 DR.INT.RATE.... 18.00
  8. 1 DR.MARGIN.OPER.
  9. 1 DR.MIN.RATE.... 12.00
 10. 1 DR.MARGIN.RATE.
 11. 1 DR.LIMIT.AMT...
 12. 1 DR.MAX.RATE....
 13 DR2.BALANCE.TYPE..
 14 DR2.CALCUL.TYPE...
 15. 1 DR2.BASIC.RATE.
 16. 1 DR2.INT.RATE...
------------------------------------------------------------------------------
16 DEC 2014 17:42:06  USER (22 APR) INPUTTER          [8572,I PAGE 1   >>>3>>>
ACTION
```

Now the jQL query gives us:

jsh

```
LIST FBNK.ACCOUNT '66044' ADI.ID
```

Output

```
@ID.............    ADI.ID..................

              66044    66044-20140324
                       66044-20140422
                       66044-20150101
```

*(Note that SYS.SINGLE.MULT in SS record is set to M - multi-valued - and therefore returned values are displayed accordingly to that setting.)*

## 1.59 J-descriptors

J-descriptors as such don’t exist at DBMS level; they are actually I-descriptors that call the standard subroutine to retrieve data from another table, thus serving like “left join” in SQL.

Example:

T24

```
 Model Bank R14          STANDARD SELECTION FIELDS SEE

     FILE.NAME......... ACCOUNT
 ------------------------------------------------------------------------------
   1. 9 SYS.FIELD.NAME. SECTOR
   2. 9 SYS.TYPE....... J
   3. 9. 1 SYS.FIELD.NO CUSTOMER.NO>CUSTOMER>SECTOR
   6. 9 SYS.DISPLAY.FMT 4R
  10. 9 SYS.SINGLE.MULT S
   1.10 SYS.FIELD.NAME. INDUSTRY
   2.10 SYS.TYPE....... J
   3.10. 1 SYS.FIELD.NO CUSTOMER.NO>CUSTOMER>INDUSTRY
   6.10 SYS.DISPLAY.FMT 4R
  10.10 SYS.SINGLE.MULT S
   1.11 SYS.FIELD.NAME. TARGET
   2.11 SYS.TYPE....... J
   3.11. 1 SYS.FIELD.NO CUSTOMER.NO>CUSTOMER>TARGET
   6.11 SYS.DISPLAY.FMT 4R
  10.11 SYS.SINGLE.MULT S
   1.12 SYS.FIELD.NAME. RESIDENCE
 ------------------------------------------------------------------------------
 16 DEC 2014 16:54:45  USER (22 APR) INPUTTER          [19745,IPAGE 4   >>137>>
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

We already made selects from INDUSTRY and TARGET which are used for classification of customers.  SECTOR is another one:

T24

```
Model Bank R14

     Id  Description
------------------------------------------------------------------------------
 1  1000  ***Individuals**
 2  1001  Individual
 3  1002  Staff
 4  1499  ***Individuals
 5  1500  ***Brokers
 6  1501  Broker
 7  1599  ***Brokers
 8  1600  ***Clearing Agents
 9  1601  Clearing Agents
10  1699  ***Clearing Agents
11  1900  ***Other Individuals
12  1999  ***Other Individuals
13  2000  ***Corporate
14  2001  Corporate
15  2002  Unincorporated Businesses
16  2003  Subsidiary Company
17--2005--Retail Small/Med Enterprise       ----------------------------------
186 299920***Corporate                              [26739,IPAGE   1 >>>1>>>
19CT3000  ***Banks
4AWAITING PAGE INSTRUCTIONS
```

...again screen corruption that we’re going to correct:

T24

```
Model Bank R14          ENQUIRY, INPUT

    ENQUIRY........... %SECTOR
------------------------------------------------------------------------------
  1 PAGE.SIZE ........ 4,19                                    <== was: 4,99
  2 FILE.NAME......... SECTOR
  3. 1 FIXE
  4. 1 FIXED.SORT.....
  5. 1 OPEN.BRACKET...
  6. 1 SELECTION.FLDS.
  7. 1. 1 GB SEL.LABEL
  8. 1 SEL.FLD.OPER...
  9. 1 REQUIRED.SEL...
 10. 1 CLOSE.BRACKET..
 11. 1 REL.NEXT.FIELD.
 12. 1 BUILD.ROUTINE..
 13. 1. 1 HEADER...... @(6,2)Id
 13. 1. 2 HEADER...... @(10,2)Description
 14. 1 FIELD.NAME..... @ID
 15. 1. 1 OPERATION... @ID
------------------------------------------------------------------------------
16 DEC 2014 17:54:18  USER (22 APR) INPUTTER          [26739,IPAGE 1   >>>7>>>
ACTION
```

Relogin to T24 and:

T24

```
 Model Bank R14

      Id  Description
 ------------------------------------------------------------------------------
 1  1000  ***Individuals**
 2  1001  Individual
 3  1002  Staff
 4  1499  ***Individuals
 5  1500  ***Brokers
 6  1501  Broker
 7  1599  ***Brokers
 8  1600  ***Clearing Agents
 9  1601  Clearing Agents
10  1699  ***Clearing Agents
11  1900  ***Other Individuals
12  1999  ***Other Individuals
13  2000  ***Corporate
14  2001  Corporate
15  2002  Unincorporated Businesses
16  2003  Subsidiary Company
 ------------------------------------------------------------------------------
 16 DEC 2014 17:56:55  USER (22 APR) INPUTTER          [9146,INPAGE   1 >>>3>>>
 ACTION _
 AWAITING PAGE INSTRUCTIONS
```

Sector @ID is attached to CUSTOMER in the data field:

jsh

```
CT DICT FBNK.CUSTOMER SECTOR
```

Output

```
    SECTOR
001 D
002 23
003
004 SECTOR
005 4R
006 S
007
...
```

jsh

```
LIST FBNK.CUSTOMER SECTOR
```

Output

```
@ID.......    SECTOR

    188888      1501
    122122      1002
    100362      3503
    120110      2005
    129003      1001
    ...
```

In ACCOUNT SECTOR is a calculated field:

jsh

```
CT DICT FBNK.ACCOUNT SECTOR
```

Output

```
        SECTOR
001 I
002 CUSTOMER.NO; SUBR("ENQ.TRANS","CUSTOMER", @1, "SECTOR")
003
004 SECTOR
005 4R
006 S
007
```

However we can list it directly or use in T24 enquiry:

jsh

```
LIST FBNK.ACCOUNT SECTOR
```

Output

```
@ID................    SECTOR

              70017      2005
              14095      3501
   USD1405502940001
              14087      3501
              67997      1001
       USD140300001
   USD1590599990001
              71676      1001
              72206      1001
       EUR150050001
   ...
```

Again, in a “fresh” session (without prior login to T24) the output includes error messages:

jsh

```
@ID.................    SECTOR

              70017        Un
                          able t
                          o open
                           F0.CU
                          STOMER
              14095        Un
                          able t
                          o open
                           F0.CU
                          STOMER
   USD1405502940001        Un
                          able t
                          o open
                           F0.CU
                          STOMER
              14087        Un
                          able t
                          o open
                           F0.CU
                          STOMER
    ...
```

For internal accounts like USD140300001 SECTOR is empty because there’s no CUSTOMER attached to them:

jsh

```
LIST FBNK.ACCOUNT CUSTOMER SECTOR
```

Output

```
@ID.................    CUSTOMER..    SECTOR

              70017        100310      2005
              14095        100394      3501
   USD1405502940001
              14087        100394      3501
              67997        100391      1001
       USD140300001
   USD1590599990001
              71676        100802      1001
              72206        100100      1001
       EUR150050001
  ...
```

We can also use T correlative to list this data in jQL without being dependent on T24:

jsh

```
LIST FBNK.ACCOUNT EVAL "OCONV(CUSTOMER,'TFBNK.CUSTOMER;V1;23;23')"
```

*(23 is the number of SECTOR field in CUSTOMER table.)*

Output

```
@ID.............    OCONV(CUSTOMER,"TFBNK.CUSTOMER;V1;23;23")

              70017                                          2005
              14095                                          3501
   USD1405502940001
              14087                                          3501
              67997                                          1001
       USD140300001
   USD1590599990001
              71676                                          1001
              72206                                          1001
       EUR150050001
              70262                                          1001
...
```

To decrease the width of second column we can either give it an alias...

jsh

```
LIST FBNK.ACCOUNT EVAL "OCONV(CUSTOMER,'TFBNK.CUSTOMER;V1;23;23')" AS Sector
```

Output

```
@ID.............    Sector....

              70017          2005
              14095          3501
   USD1405502940001
              14087          3501
              67997          1001
       USD140300001
   USD1590599990001
              71676          1001
              72206          1001
       EUR150050001
              70262          1001
...
```

...or assign column headers:

jsh

```
LIST FBNK.ACCOUNT ID.SUPP @ID COL.HDR 'Account No' EVAL "OCONV(CUSTOMER,
   'TFBNK.CUSTOMER;V1;23;23')" COL.HDR 'Customer sector'
```

Output

```
Account No.........    Customer sector

              70017               2005
              14095               3501
   USD1405502940001
              14087               3501
              67997               1001
       USD140300001
   USD1590599990001
              71676               1001
              72206               1001
       EUR150050001
              70262               1001
...
```

Finally, yet another alternative to J-descriptor is XLATE() function.  The following jQL query gives the same result as is shown at the previous screen:

jsh

```
LIST FBNK.ACCOUNT ID.SUPP @ID COL.HDR 'Account No' EVAL "XLATE('FBNK.CUSTOMER',
   CUSTOMER, 23, 'X')" COL.HDR 'Customer sector'
```

## 1.60  More output options in jQL

Let’s sort the output shown before the last example in the previous chapter (and add the filter to list only accounts with sectors, i.e.  customer accounts):

jsh

```
LIST FBNK.ACCOUNT ID.SUPP @ID COL.HDR 'Account No' EVAL "OCONV(CUSTOMER,
   'TFBNK.CUSTOMER;V1;23;23')" AS Sect COL.HDR 'Customer sector'
   WITH Sect NE '' BY Sect
```

Output

```
Account No.........    Customer sector

              10944               1001
              10968               1001
              10979               1001
              11298               1001
              11312               1001
              11328               1001
              11495               1001
...
```

Add breaks whenever customer sector changes using keyword BREAK.ON:

jsh

```
LIST FBNK.ACCOUNT ID.SUPP @ID COL.HDR 'Account No' BREAK.ON EVAL "OCONV(CUSTOMER,
   'TFBNK.CUSTOMER;V1;23;23')" AS Sect WITH Sect NE '' BY Sect
```

Output

```
 BY Sect PAGE   18 22:52:01  16 DEC 2014

Account No.........    Sect......

              72427          1001
              72435          1001
              72467          1001
              72478          1001
              72486          1001
              72494          1001
              72575          1001

                              ***


              22098          1002
              22101          1002
              66354          1002
              69248          1002
              69817          1002
              70475          1002
              71258          1002

                              ***
...
```

Add sequential number to the output (*A9998):

jsh

```
LIST FBNK.ACCOUNT ID.SUPP *A9998 COL.HDR '#' @ID COL.HDR 'Account No' BREAK.ON EVAL
   "OCONV(CUSTOMER,'TFBNK.CUSTOMER;V1;23;23')" AS Sect WITH Sect NE '' BY Sect
```

Output

```
#.............    Account No.........    Sect......

             1                  10944          1001
             2                  10968          1001
             3                  10979          1001
             4                  11298          1001
             5                  11312          1001
             6                  11328          1001
             7                  11495          1001
...
```

Calculate total WORKING.BALANCE for a sector.  Since account currencies are different, add currency to sort and to breaks:

jsh

```
LIST FBNK.ACCOUNT ID.SUPP @ID COL.HDR 'Account No' TOTAL WORKING.BALANCE BREAK.ON
   EVAL "OCONV(CUSTOMER,'TFBNK.CUSTOMER;V1;23;23')" AS Sector BREAK.ON CURRENCY
   WITH Sector NE '' BY Sector BY CURRENCY
```

Output

```
Account No.........    WORKING.BALANCE....    Sector....    CURRENCY

              69019               62677.01          1001    CAD
              69078               96242.03          1001    CAD

                                 158919.04                  ***


              11509               -1689.95          1001    CHF
              14378               23213.47          1001    CHF
              67083                                 1001    CHF
              67091                                 1001    CHF

                                  21523.52                  ***
...
              72494                                 1001    USD
              72575               -51612.8          1001    USD

                                 4169772.9                  ***


                                3764567.13           ***

              22101              -22841.72          1002    EUR

                                 -22841.72                  ***
...
```

Suppress details, leaving only totals - DET-SUPP keyword (account no more makes sense so get rid of it as well):

jsh

```
LIST FBNK.ACCOUNT ID.SUPP TOTAL WORKING.BALANCE BREAK.ON EVAL "OCONV(CUSTOMER,
   'TFBNK.CUSTOMER;V1;23;23')" AS Sector BREAK.ON CURRENCY WITH Sector NE ''
   BY Sector BY CURRENCY DET-SUPP
```

Output

```
WORKING.BALANCE....    Sector....    CURRENCY

          158919.04                  CAD
           21523.52                  CHF
          180652.42                  EUR
         -836230.75                  GBP
                  0                  JPY
              69930                  SGD
           4169772.9                 USD

         3764567.13          1001
          -22841.72                  EUR
         -505802.59                  USD

         -528644.31          1002
         -720341.55                  EUR

         -720341.55          1499
                  0                  EUR
             359544                  USD

             359544          1501
...
          377459.67          3504
              -5035                  EUR
                  0                  GBP
            -164.45                  USD

           -5199.45          3505
====================
       304285749.69

 758 Records Listed
```

Reverse sort can be achieved using BY.DSND keyword:

jsh

```
LIST FBNK.ACCOUNT ID.SUPP TOTAL WORKING.BALANCE BREAK.ON EVAL "OCONV(CUSTOMER,
   'TFBNK.CUSTOMER;V1;23;23')" AS Sector BREAK.ON CURRENCY WITH Sector NE ''
   BY.DSND Sector BY CURRENCY DET-SUPP
```

Output

```
WORKING.BALANCE....    Sector....    CURRENCY

              -5035                  EUR
                  0                  GBP
            -164.45                  USD

           -5199.45          3505
          377459.67                  USD

          377459.67          3504
          -63673.55                  EUR
            -113.78                  GBP
          265836.34                  USD
...
```

## 1.61 CATEGORY

Another important application in T24 is CATEGORY - full list of all bank products:

T24

```
 Model Bank R14               Category list

   Id      Description                              Short Name       Prod
 ------------------------------------------------------------------------------
 1    1000 ***Demand Accounts                       Demand Acct      AC
 2    1001 Current Account                          Current Account  AC
 3    1002 Current Account with Overdraft           CurrAcc with OD  AC
 4    1003 Current Account (Premium)                Premium C/A      AC
 5    1004 Current Account (Student)                Student C/A      AC
 6    1005 Export Packing Credit                    Packing Credit   AC
 7    1006 Call 2                                   Call 2           AC
 8    1007 Prefer Current Account                   Prefer Current   AC
 9    1010 Credit Cards                             Credit Card      AC
10    1999 ***Demand Accounts                       Demand Acct      AC
11    2000 *** Vostro Accounts                      Vostro Acct      AC
12    2001 Vostro Accounts                          Vostro Accounts  AC
13    2999 *** Vostro Accounts                      Vostro Acct      AC
14    3000 *** Loan Accounts                        Loan Accts       AC
15    3001 Mortgage Account                         Mortgage Acct    AC
16    3101 AA Commercial Loan                       AA Comm. Loan    AC
 ------------------------------------------------------------------------------
 16 DEC 2014 18:06:07  USER (22 APR) INPUTTER          [6290,INPAGE   1 >>>3>>>
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

All customer and internal accounts, loans, cash tills, ATMs, letters of credit, guarantees etc should have a category attached:

jsh

```
LIST FBNK.ACCOUNT CATEGORY EVAL "OCONV(CATEGORY,'TF.CATEGORY;V1;1;1')"
   COL.HDR 'Full product name from CATEGORY table'
```

Output

```
@ID..............    CATEGORY    Full product name from CATEGORY table

              70017        3104                        AA Negotiable Loan
              14095        1001                           Current Account
   USD1405502940001       14055                    Daily Difference in DC
              14087        1001                           Current Account
              67997        1001                           Current Account
       USD140300001       14030                    Inw Clg Debit Suspense
   USD1590599990001       15905                           TC Stock Contra
              71676        3102                         AA Personal Loan
              72206        3103                              AA Mortgage
       EUR150050001       15005                          Managers Checks
              70262        3101                        AA Commercial Loan
...
```

*(Category @ID for internal account is duplicated in its @ID.)*

Other financial applications:

jsh

```
LIST FBNK.LD.LOANS.AND.DEPOSITS CATEGORY EVAL "OCONV(CATEGORY,'TF.CATEGORY;V1;1;1')"
   COL.HDR 'Full product name from CATEGORY table'
```

Output

```
@ID.....................  CATEGORY  Full product name from CATEGORY table

             LD1409811078     21096            Commitment (Loan) Contracts
             LD1408377464     21051                          Annuity Loan
             LD1411292330     21051                          Annuity Loan
             LD1408790756     21054                     Construction Loan
             LD1408389709     21053                          Project Loan
             LD1408391042     21052                           Bridge Loan
             LD1408302938     21053                          Project Loan
             LD1408382705     21052                           Bridge Loan
             LD1408305044     21003                          Term Deposit
             ...
```

jsh

```
LIST FBNK.MM.MONEY.MARKET CATEGORY EVAL "OCONV(CATEGORY,'TF.CATEGORY;V1;1;1')"
   COL.HDR 'Full product name from CATEGORY table'
```

Output

```
@ID..........    CATEGORY    Full product name from CATEGORY table

  MM1408300061       21077                     Loans to Banks (MM)
  MM1408300025       21032                  Deposits from Banks(MM)
  MM1408700009       21002                Interest Bearing Deposits
  MM1411200023       21002                Interest Bearing Deposits
  MM1408300062       21002                Interest Bearing Deposits
  MM1409700017       21076                   Call/Notice Placements
  MM1409700018       21077                     Loans to Banks (MM)
  MM1408700006       21002                Interest Bearing Deposits
  MM1408300058       21031                    Call / Notice Takings
  MM1411200009       21200                                ***Repo
...
```

jsh

```
LIST FBNK.MD.DEAL CATEGORY EVAL "OCONV(CATEGORY,'TF.CATEGORY;V1;1;1')"
   COL.HDR 'Full product name from CATEGORY table'
```

Output

```
@ID.............    CATEGORY    Full product name from CATEGORY table

MD1408707495            28102            Guarantee Received - Generic
MD1409806936            28023          Import Lc - Shipping Guarantee
MD1409133059            28023          Import Lc - Shipping Guarantee
MD1408389012            28001                       Generic Guarantee
MD1411233445            28015                     Standby Guarantees
MD1409769512            28001                       Generic Guarantee
MD1411206400            28010                  Performance Guarantee
MD1408300421            28005                     Bid Bond Guarantee
...
```

Categories are divided into ranges (e.g.  for customer accounts - 1000 to 9999, for internal accounts - 10000 to 19999 etc).  There are also so-calleed “P&L” categories where incomes, expenses, profits and losses are booked (@IDs starting from 50000).

## 1.62  Other T24 application types

We already studied some application types:

- H: application with NAU and history.
- U: application with NAU but without history.
- W: application without NAU and history; “V” function can be used to trigger a special action.

Other T24 application types are:

- L: readonly application that doesn’t allow input in T24 (has only one “LIVE” data file; populated only in a program or at jsh level using COPY or JED).
- T: concat files like CUSTOMER.ACCOUNT. Also doesn’a allow editing at T24 level; is populated by T24 core.
- D: dynamic ones - can be created without programming.

Examples:

jsh

```
LIST F.PGM.FILE 'ACCT.ACTIVITY' TYPE
```

Output

```
@ID.....................................    TYPE

ACCT.ACTIVITY                               L
```

No input (ACCT.ACTIVITY I):

T24

```
 Model Bank R14          Account Activity


 ------------------------------------------------------------------------------









 -- LAST SIGN.ON, DATE: 02 FEB 2015     TIME: 14:36        ATTEMPTS: 0 --------
 03 FEB 2015 16:52:14  USER (22 APR) VLADIMIR.K        [26400,IN]
 ACTION I                                          INVALID FUNCTION FOR LIVE-FILE
 AWAITING FUNCTION                                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

No INAU file:

T24

```
-- LAST SIGN.ON, DATE: 31 MAR 2015     TIME: 11:04        ATTEMPTS: 0 --------
31 MAR 2015 11:50:45  USER (22 APR) VLADIMIR.K        [32037,IN]
ACTION E                                          INVALID FUNCTION FOR LIVE-FILE
AWAITING FUNCTION                                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

No history (ACCT.ACTIVITY L ; L):

T24

```
Model Bank R14          Account Activity LIST


--------------------------------------------------------------------------------
       SELECTION FIELDS - LIVE FILE      [EQ NE LT GT LE GE RG NR LK UL]
--------------------------------------------------------------------------------
0.. @ID                    0A. ACCT.NO.YEAR.MONTH    1.. DAY.NO
2.. TURNOVER.CREDIT        3.. TURNOVER.DEBIT        4.. BALANCE
5.. TRANSACT.CODE          6.. NO.OF.TRANSACT        7.. TRANSACT.AMT
8.. VALUE.DAY              9.. BOOKING.DATE          10. BOOK.TOVR.CR
11. BOOK.TOVR.DB           12. TRADE.DATE            13. TRADE.TOVR.CR
14. TRADE.TOVR.DB          15. BK.DAY.NO             16. BK.BALANCE
17. BK.CREDIT.MVMT         18. BK.DEBIT.MVMT         19. BOOKING.DAY
20. BK.TRADE.DATE          21. BK.TRADE.TOVR.CR      22. BK.TRADE.TOVR.DB







--------------------------------------------------------------------------------
31 MAR 2015 11:52:35  USER (22 APR) VLADIMIR.K        [32037,INPAGE  1 >>1>>>
ACTION                                                  EB.RTN.NO.HIS.FILE.APP
ENTER SELECTION (eg CUST EQ 123456) , <F5> TO EXECUTE LIST  ^^^^^^^^^^^^^^^^^^^^^^
```

Same for T type:

jsh

```
LIST F.PGM.FILE 'CUSTOMER.ACCOUNT' TYPE
```

Output

```
@ID.....................................    TYPE

CUSTOMER.ACCOUNT                           T
```

T24 (CUSTOMER.ACCOUNT D)

```
--------------------------------------------------------------------------------
31 MAR 2015 11:54:31  USER (22 APR) VLADIMIR.K        [32037,IN]
ACTION D                                          INVALID FUNCTION FOR TABLE FILE
AWAITING FUNCTION                                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

## 1.63 T24 transactions

In local developments it’s strictly forbidden to deal with transactions. The exception is call to JOURNAL.UPDATE in “mainline” routine (PGM.FILE type “M”) but even in this case we don’t need to declare the transaction explicitly. Another exception is “W” template where the same call can be used upon “V” function invocation.

We can check whether we are inside a transaction:

jBC

```
CRT SYSTEM(47)    ;* 1 - we're inside a transaction, 0 - we're not
```

For example, in ID.RTN which is another VERSION hook we’re not yet in T24 transaction. To illustrate this (and the fact that we can amend the @ID that was inputted by the user) we’ll append the result of SYSTEM(47) to the @ID (which of course doesn’t have any practical usage).

jBC

```
SUBROUTINE SUBR.IR
* T24 ID routine
$INSERT I_COMMON
$INSERT I_EQUATE

DEBUG
   COMI := SYSTEM(47)

   RETURN
END
```

Trying to attach it to VERSION AB,TRAIN...

T24

```
 Model Bank R14          VERSION, INPUT

     PGM.NAME.VERSION.. ABBREVIATION,TRAIN
 ------------------------------------------------------------------------------
  65 REPORT.LOCKS...... YES
  66 GTS.CONTROL.......
  67. 1 GB D
  68. 1 ASSOC.VERSION..
  69 NEXT.VE
  70 INITIAL.CURSOR.POS
  71. 1 RESERVED.4.....
  72. 1 CAPTION........
  73 EXC.INC.RTN.......
  74. 1 ID.RTN......... SUBR.IR                              RECORD NOT FOUND
  75. 1 CHECK.REC.RTN..
  76. 1 AFTER.UNAU.RTN.
  77. 1 BEFORE.AUTH.RTN
  78 VERSION.TYPE......
  79 ENABLE.GRID.......
  80 MAX.DISPLAY.LINES.
 ------------------------------------------------------------------------------
```

“RECORD NOT FOUND” isn’t very informative but it’s a usual F.READ reply when the record couldn’t be read.  We can guess that it’s EB.API record that the core was looking for.

After all necessary actions, run a VERSION: AB,TRAIN I QWE. We can see that at the stage of @ID input T24 transaction hadn’t yet started and that ID.NEW is still empty. We use the global variable COMI to get and to set the @ID:

Debugger

```
 ------------------------------------------------------------------------------
DEBUG statement seen
Source changed to D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\ETC.BP\SUBR.IR
0005     DEBUG
jBASE debugger->V ID.NEW
COMMON variables
  ID.NEW                    :
jBASE debugger->V COMI
COMMON variables
  COMI                      : QWE
```

When we type C to continue execution we’ll stop at another DEBUG statement that presents in field auto-population routine SUBR.ANC. Now ID.NEW is QWE0:

Debugger

```
jBASE debugger->C
DEBUG statement seen
Source changed to D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\ETC.BP\SUBR.ANC
0004 DEBUG
jBASE debugger->V ID.NEW
COMMON variables
  ID.NEW                    : QWE0
```

## 1.64 F.PROTOCOL

We can see every action of a user in F.PROTOCOL table provided that user has the following fields set to Y:

T24

```
Model Bank R14          USER PROFILE SEE

    USER.ID........... INPUTTER
------------------------------------------------------------------------------
   ...
     25 SIGN.ON.OFF.LOG... NO
     26 SECURITY.MGMT.L... NO
     27 APPLICATION.LOG... NO
     28 FUNCTION.ID.LOG... NO
   ...
```

Let’s set them and see the impact after some actions in T24.

jsh

```
LIST F.PROTOCOL
```

Output

```
201406052022554279.03
201406052022554286.00
201406052022554319.00
201406065045827247.02
201406067565727660.00
201406066332427679.02
```

jsh

```
LIST DICT F.PROTOCOL
```

Output

```
  No Records Listed
```

Again, no dictionary...  Let’s try to correct that by T24 command SS I PROTOCOL (there’s no “comma” VERSION and this time we’ll authorise it right after input). Set the field REBUILD.SYS.FLDS to Y and commit, then try to authorise:

T24

```
 ------------------------------------------------------------------------------
 16 DEC 2014 20:15:09  USER (22 APR) INPUTTER          [10124,
 ACTION PROTOCOL                          EB.RTN.SAME.NAME.AUTHORISER/INPUTTER
 AWAITING ID
```

Authorisation is to be done by another user:

T24

```
Model Bank R14          STANDARD SELECTION FIELDS AUTHORISE

    FILE.NAME......... PROTOCOL
------------------------------------------------------------------------------
 12.16 SYS.GENERATED.. Y
 28 REBUILD.SYS.FLDS.. Y

 34 RECORD.STATUS..... INAU                INPUT Non validé
 35 CURR.NO........... 7
 36. 1 INPUTTER....... 2052_INPUTTER
 37. 1 DATE.TIME...... 16 DEC 14 20:08
 39 CO.CODE........... GB-001-0001         Model Bank R14
 40 DEPT.CODE......... 1                   Implementation









------------------------------------------------------------------------------
16 DEC 2014 20:09:35  USER (22 APR) VLADIMIR.K1       [10846,IPAGE 10
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Now we have the dictionary...

jsh

```
LIST DICT F.PROTOCOL
```

Output

```
TERMINAL.ID
COMPANY.ID
REMARK
PROTOCOL@XMLMAP
@ID
PROTOCOL.ID
@
XBUILD.DATEX
...
```

...so we can list the protocol for the user INPUTTER for today (system date is the @ID start):

jsh

```
LIST F.PROTOCOL WITH @ID LIKE "'20141216'..." AND USER EQ 'INPUTTER'
```

Output

```
@ID.............. 201412161012472902.02
@ID.............. 201412161012472902.02
PROTOCOL.ID....... 201412161012472902.02
PROCESS.DATE...... 20140422
DATE.VERSION...... 1
TIME.............. 201502
TIME.MSECS........ 20:15:02:405
TERMINAL.ID....... 10124
PHANTOM.ID........
COMPANY.ID........ GB0010001
USER.............. INPUTTER
APPLICATION....... STANDARD.SELECTION
LEVEL.FUNCTION.... 1 I
ID................ PROTOCOL
REMARK............
CLIENT.IP.ADDRESS.
LOCAL.DATE.TIME... 141216201502405
...
```

## 1.65 Financial transactions

We learned already about T24 setup and static tables like SPF, COMPANY, CATEGORY, INDUSTRY etc. Financial transactions (like money transfers) are entered on day-to-day basis and at the end of business day are moved to history (though it’s not the global rule).

We already tried to amend the FUNDS.TRANSFER transaction in OFS. Example of another application where financial transactions are stored is TELLER:

T24

```
Model Bank R14          TELLER SEE

    TRANSACTION.NUMBER TT/14112/3BZ1Q
--------------------------------------------------------------------------------
  1 TRANSACTION.CODE.. 76                  Issue LCY Drafts - Cash
  2 TELLER.ID.1....... 1510                TELLER.1
  3 DR.CR.MARKER...... CREDIT
  4 CURRENCY.1........ USD                 US Dollar
  6. 1 ACCOUNT.1...... USD-14046-0001-0001 RECORD.AUTOMATICALLY.OPENED
  7. 1 AMOUNT.LOCAL.1. 750.00
 11 VALUE.DATE.1...... 22 APR 2014
 13 CURR.MARKET.1..... 1                   Currency Market
 14 POS.TYPE.1........ TR                  TRADING POSITION
 16 CURRENCY.2........ USD                 US Dollar
 17 TELLER.ID.2....... 1510                TELLER.1
 18 ACCOUNT.2......... USD-10001-1510-0001 RECORD.AUTOMATICALLY.OPENED
 20 AMOUNT.LOCAL.2.... 750.00
 23 NET.AMOUNT........ 755.00
 24 VALUE.DATE.2...... 22 APR 2014
 26 CURR.MARKET.2..... 1                   Currency Market
--------------------------------------------------------------------------------
```

To input teller transactions one need to have TELLER.ID assigned.  It will be easier to input new FT transaction then.  To generate the new @ID automatically press F3 at “AWAITING ID” prompt:

T24

```
Model Bank R14          FUNDS.TRANSFER INPUT                  REF FT14112951B7


--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE.. _
  2 DEBIT.ACCT.NO.....
  3 IN.DEBIT.ACCT.NO..
  4 CURRENCY.MKT.DR... 1                  Currency Market
  5 DEBIT.CURRENCY....
  6 DEBIT.AMOUNT......
  7 DEBIT.VALUE.DATE..
  8 IN.DEBIT.VDATE....
  9 DEBIT.THEIR.REF...
 10 CREDIT.THEIR.REF..
 11 CREDIT.ACCT.NO....
 12 CURRENCY.MKT.CR... 1                  Currency Market
 13 CREDIT.CURRENCY...
 14 CREDIT.AMOUNT.....
 15 CREDIT.VALUE.DATE.
 16 TREASURY.RATE.....
--------------------------------------------------------------------------------
16 DEC 2014 20:37:14  USER (22 APR) INPUTTER          [16543,IPAGE 1   >>14>>>
ACTION
```

*(We use here “pure” application - not a VERSION.)*

FT @ID structure is:  2-character application prefix, year, current day number in a year followed by the unique number for this day’s transactions.

Current day number is taken not from system date - if it was it would be...

jsh

```
LIST . EVAL "OCONV(DATE(),'DJ')" SAMPLE 1 ID-SUPP
```

Output

```
351
```

...but from current T24 working day:

jsh

```
LIST F.DATES EVAL "OCONV(ICONV(TODAY,'D'),'DJ')" SAMPLE 1
```

Output

```
@ID.............    OCONV(ICONV(TODAY,"D"),"DJ")

GB0010005                                   112
```

*(ICONV() to convert TODAY to internal jBASE format was necessary because TODAY is stored in format YYYYMMDD; DATE() output is already in internal format.)*

“LIST . EVAL ’something’ SAMPLE 1 ID-SUPP” construction can be used to test some jBC functionality; it works the same way for almost all cases, e.g.  get the absolute value of a number:

jsh

```
LIST . EVAL "ABS(-5)" SAMPLE 1 ID-SUPP
```

Output

```
ABS(-5)...

5
```

Getting absolute value of all elements of an array works in jBC but not here:

jsh

```
LIST . EVAL "ABSS(-5:@FM:-4:@VM:-3)" SAMPLE 1 ID-SUPP
```

Output

```
ABSS(-5:@FM:-4:@VM:-3)

5
0
0
```

Back to auto-assignment of @ID. How the unique number was generated?

In earlier times @IDs were assigned consequently and the corresponding record in “L” application LOCKING was used to store the counter.

Again, see that not all functions are available for “L” type:

T24 application LOCKING

```
 ------------------------------------------------------------------------------
 16 DEC 2014 20:39:19  USER (22 APR) INPUTTER          [16543,IN]
 ACTION E                                        INVALID FUNCTION FOR LIVE-FILE
 AWAITING FUNCTION
```

*(“LIVE” rather meaning “live-only”; readonly and without NAU and HIS that is L-type application - hence E function absents; same for functions D, A, R and H.)*

To access the desired record quickier, enter command L -FUNDS

T24

```
--------------------------------------------------------------------------------
16 DEC 2014 20:42:25  USER (22 APR) INPUTTER          [16543,IN]
ACTION L -FUNDS
AWAITING FUNCTION
```

...and see the list starting from “FUNDS”:

T24

```
Model Bank R14          LOCKING LIST

NO. ID     RECORD FIELDS                                                FUNCT.
--------------------------------------------------------------------------------
  1 FUNDS.TRANSFER.UNIQUE
  2 FX.BULK.ORDER.UNIQUE  0123456789
  3 FX.COMM.GROUP.UNIQUE  0123456789
  4 FX.LIM.ORDER.UNIQUE  0123456789
  5 FX.ORDER.UNIQUE  0123456789
  6 FX.TREASURY.PW.PROCESS.UNIQUE  0123456789
  7 GB0010001.NR.ITEMS  46
  8 GI
  9 GPAC.MDC.H.DATA.CAPTURE.UNIQUE  12
 10 IDS  N
 11 IM.DOCUMENT.IMAGE.UNIQUE  0123456789
 12 J28445.19980723  F.CONVERSION.PGMS G13
 13 J28446.19980723  F.CONVERSION.PGMS$NAU G14
 14 J28447.19980723  F.FILE.CONTROL MG.RATE.EXCEPTION
 15 JOURNAL  0
 16 LAST.CUSTOMER.STATIC-GB0010001  201405160523  201405160412
--------------------------------------------------------------------------------
16 DEC 2014 20:42:58  USER (22 APR) INPUTTER          [16543,IPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

*(F.LOCKING contains a lot of other records as well.)*

The record with counter was locked whenever user created a new record without entering @ID. On high load it presented a performance bottleneck.  In additon, 99,999 FTs per day also was a limit some banks couldn’t afford.

See such counter for CUSTOMER that is still used...

T24

```
Model Bank R14          LOCKING SEE

    KEY.............. FBNK.CUSTOMER
--------------------------------------------------------------------------------
  1. 1 CONTENT........ 129029
```

...increase it (CU I, then F3)...

T24

```
Model Bank R14          CUSTOMER INPUT

    CUSTOMER.CODE..... 129030
--------------------------------------------------------------------------------
  1 MNEMONIC.......... _
  2. 1 GB SHORT.NAME..
  3. 1 GB NAME.1......
  4. 1 GB NAME.2......
  5. 1 GB STREET......
  6. 1. 1 GB ADDRESS..
  7. 1 GB TOWN.COUNTRY
  8. 1 GB POST.CODE...
  9. 1 GB COUNTRY.....
 10. 1 RELATION.CODE..
 11. 1 REL.CUSTOMER...
 12. 1 REVERS.REL.CODE
 13. 1. 1 REL.DELIV.OP
 14. 1. 1 ROLE........
 15. 1. 1 ROLE.MORE.IN
 16. 1. 1 ROLE.NOTES..
--------------------------------------------------------------------------------
16 DEC 2014 20:59:45  USER (22 APR) INPUTTER          [16543,IPAGE 1   >>12>>>
ACTION
```

...then see it again...

T24

```
Model Bank R14          LOCKING SEE

    KEY............... FBNK.CUSTOMER
--------------------------------------------------------------------------------
  1. 1 CONTENT........ 129030
```

...see also an old one for FT:

T24

```
Model Bank R14         LOCKING  SEE

    KEY............... FBNK.FUNDS.TRANSFER
--------------------------------------------------------------------------------
  1. 1 CONTENT........ FT0004100011
  2. 1 REMARK......... FT9415700001
```

To improve the performance for FUNDS.TRANSFER was developed the UNIQUE @ID generation mechanism that is defined in application AUTO.ID.START:

T24

```
Model Bank R14          Auto ID Start INPUT

    KEY.............. FUNDS.TRANSFER
-------------------------------------------------------------------------------
  1. 1 DESCRIPTION.... FUNDS TRANSFER UNIQUE ID
  2. 1 APPL FUNDS.TRANSFER
  3. 1 ID.START....... FT
  4. 1 UNIQUE.NO...... YES
  5. 1 BASE.TABLE.....
  6. 1 ID.LENGTH......
  7 RESERVED10........
  8 RESERVED9.........
  9 RESERVED8.........
 10 RESERVED7.........
 11 RESERVED6.........
 12 RESERVED5.........
 13 RESERVED4.........
 14 RESERVED3.........
 15 RESERVED2.........
 16 RESERVED1.........
-------------------------------------------------------------------------------
16 DEC 2014 20:48:56  USER (22 APR) INPUTTER          [16543, PAGE 1   >>>2>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

In COMPANY record it’s been set up which applications have automatic @ID generation (field PGM.AUTOM.ID):

T24

```
Model Bank R14          COMPANY SEE

    COMPANY.CODE...... GB-001-0001         Great Britain   COMPANY GROUP 1
-------------------------------------------------------------------------------
  1. 1 GB COMPANY.NAME Model Bank R14
  2. 1 NAME.ADDRESS... 18 Place De Philosophes,
  2. 2 NAME.ADDRESS... CH 1205 Geneva,
  2. 3 NAME.ADDRESS... Switzerland
  3 MNEMONIC.......... BNK
  4 LANGUAGE.CODE..... 1                   English
 11 CONSOLIDATION.MARK N
 12 DEFAULT.NO.OF.AUTH 1
 13. 1 PGM.AUTOM.ID... EB.ONBOARD.APPLICATION
 13. 2 PGM.AUTOM.ID... AA.ACCOUNT.CLOSURE.DETAILS
 13. 3 PGM.AUTOM.ID... AA.ARRANGEMENT.ACTIVITY
 13. 4 PGM.AUTOM.ID... AA.SIMULATION.CAPTURE
 13. 5 PGM.AUTOM.ID... AA.SIMULATION.RUNNER
 13. 6 PGM.AUTOM.ID... AC.ACC.OPENING
 13. 7 PGM.AUTOM.ID... AC.ACCOUNT.LINK
 13. 8 PGM.AUTOM.ID... AC.ACCOUNT.OPENING
-------------------------------------------------------------------------------
```

FUNDS.TRANSFER is among them:

T24 - COMPANY record

```
13.79 PGM.AUTOM.ID... FUNDS.TRANSFER
```

If application name is followed by M after space, e.g. “TELLER M”, then users have to always generate new @ID without a possibility to enter it manually (user tried to enter 1 as transaction number in the following example):

T24 - TELLER I 1

```
16 DEC 2014 21:07:23  USER (22 APR) INPUTTER          [26776,IN]
ACTION 1                                        NEW RECORD BY AUTOM. ID ONLY
```

What happens when some process holds the lock on F.LOCKING record for a long time?

jsh

```
JED F.LOCKING FBNK.CUSTOMER
```

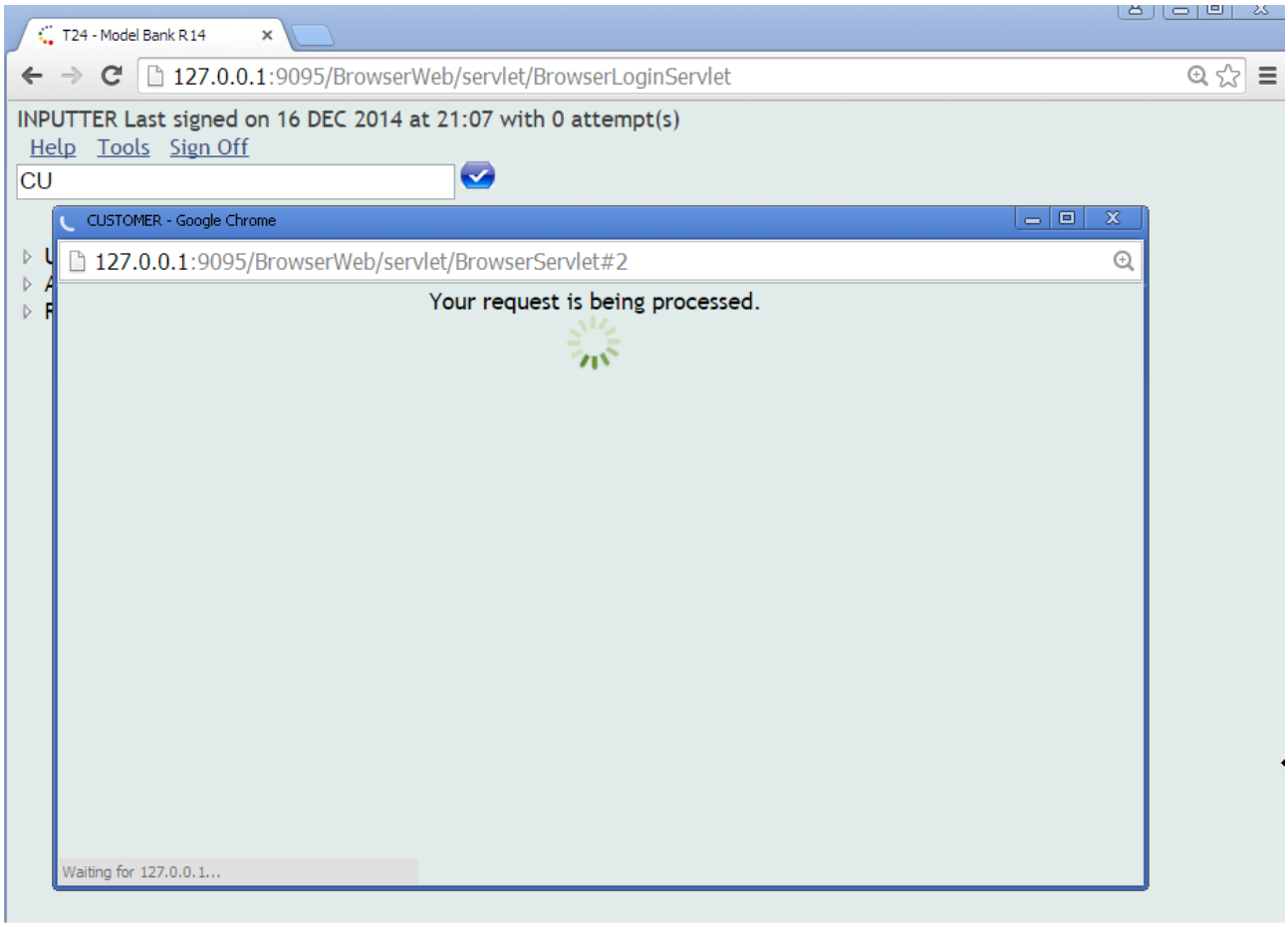


**Figure 40:  New deal - waiting.**

...wait some time, then unlock the record by exiting the JED...

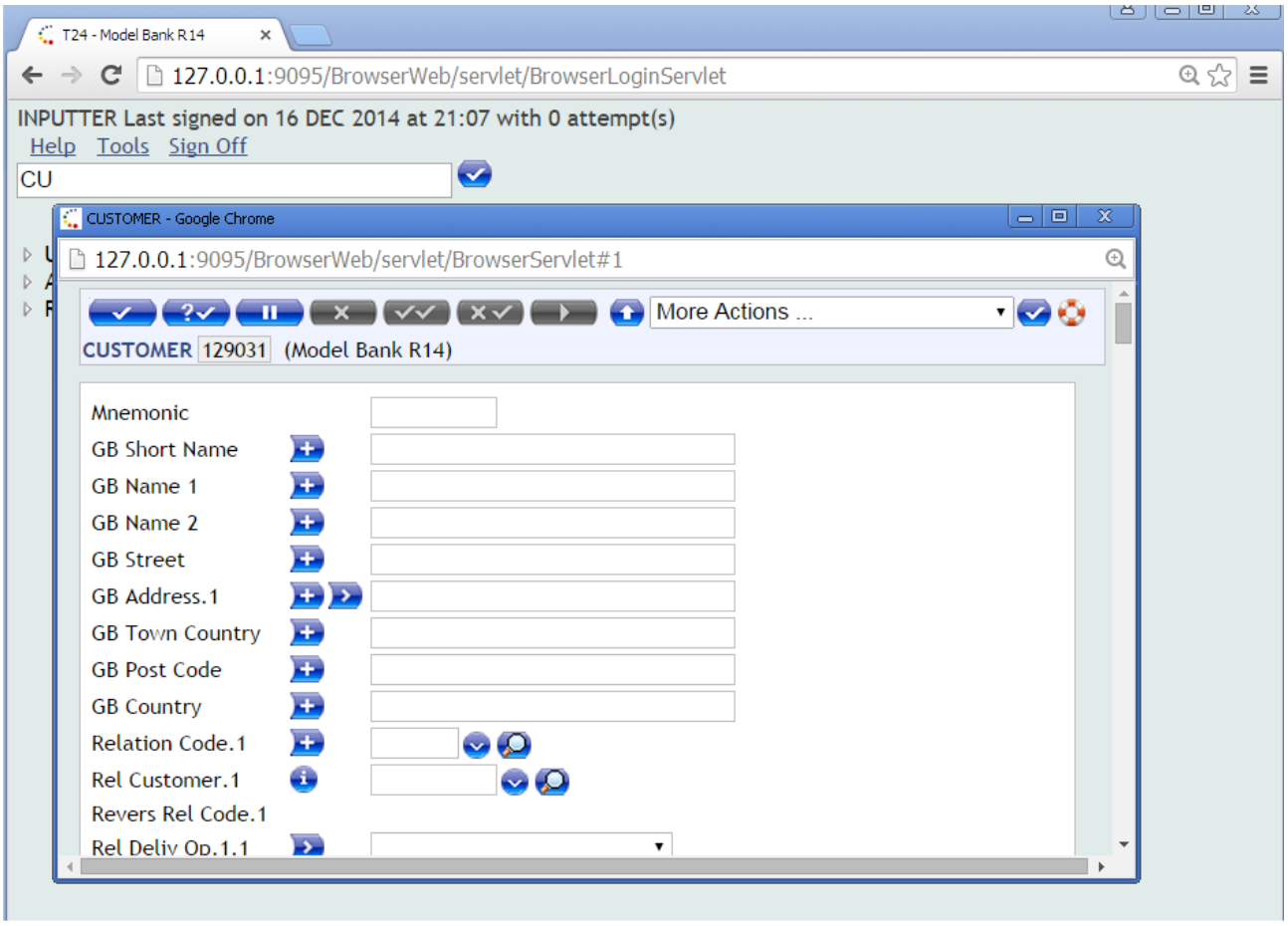


**Figure 41:  New deal - @ID obtained.**

FT doesn’t have this limitation since the unique number for the day is random. Let’s input the simplest FT - accounts transfer (in “pure” application):

T24

```
Model Bank R14          FUNDS.TRANSFER INPUT                 REF FT14112VCB74

--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE.. AC                  Account Transfer
  2 DEBIT.ACCT.NO..... 14378               MICK ROSS
  3 IN.DEBIT.ACCT.NO..
  4 CURRENCY.MKT.DR... 1                   Currency Market
  5 DEBIT.CURRENCY.... CHF                 Swiss Franc
  6 DEBIT.AMOUNT...... 30.00
  7 DEBIT.VALUE.DATE.. 22 APR 2014
  8 IN.DEBIT.VDATE....
  9 DEBIT.THEIR.REF...
 10 CREDIT.THEIR.REF..
 11 CREDIT.ACCT.NO.... 66419               Acorn Engineering Company
 12 CURRENCY.MKT.CR... 1                   Currency Market
```

See the record after commit:

T24

```
Model Bank R14          FUNDS.TRANSFER SEE                    REF FT14112VCB74


--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE.. AC                  Account Transfer
  2 DEBIT.ACCT.NO..... 14378               MICK ROSS
  4 CURRENCY.MKT.DR... 1                   Currency Market
  5 DEBIT.CURRENCY.... CHF                 Swiss Franc
  6 DEBIT.AMOUNT...... 30.00
  7 DEBIT.VALUE.DATE.. 22 APR 2014
 11 CREDIT.ACCT.NO.... 66419               Acorn Engineering Company
 12 CURRENCY.MKT.CR... 1                   Currency Market
 13 CREDIT.CURRENCY... CHF                 Swiss Franc
 15 CREDIT.VALUE.DATE. 22 APR 2014
 18 PROCESSING.DATE... 22 APR 2014
 44 CHARGE.COM.DISPLAY
 45 COMMISSION.CODE... DEBIT PLUS CHARGES
 49 CHARGE.CODE....... DEBIT PLUS CHARGES
 55 PROFIT.CENTRE.CUST 100285              Mick Ross
 57 RETURN.TO.DEPT.... NO
 62 FED.FUNDS......... NO
 63 POSITION.TYPE..... TR                  TRADING POSITION
 70 AMOUNT.DEBITED.... CHF 30.00
 71 AMOUNT.CREDITED... CHF 30.00
 81 CREDIT.COMP.CODE.. GB-001-0001         Model Bank R14
 82 DEBIT.COMP.CODE... GB-001-0001         Model Bank R14
 87 LOC.AMT.DEBITED... 33.93
 88 LOC.AMT.CREDITED.. 33.93
 94 CUST.GROUP.LEVEL.. 99
 95 DEBIT.CUSTOMER.... 100285              Mick Ross
 96 CREDIT.CUSTOMER... 128053              Acorn Engineering Company
 98 DR.ADVICE.REQD.Y.N N
 99 CR.ADVICE.REQD.Y.N N
101 CHARGED.CUSTOMER.. 128053              Acorn Engineering Company
111 TOT.REC.COMM...... 0
112 TOT.REC.COMM.LCL.. 0.00
113 TOT.REC.CHG....... 0
114 TOT.REC.CHG.LCL... 0.00
136 RATE.FIXING....... NO
148 TOT.REC.CHG.CRCCY. 0.00
195. 1 STMT.NOS....... VAL
197 RECORD.STATUS..... INAU                INPUT Unauthorised
198 CURR.NO........... 1
199. 1 INPUTTER....... 29528_INPUTTER
200. 1 DATE.TIME...... 16 DEC 14 21:22
202 CO.CODE........... GB-001-0001         Model Bank R14
203 DEPT.CODE......... 1                   Implementation
--------------------------------------------------------------------------------
```

Many core fields were populated automatically.  Record status is INAU.

Authorise it by another user (FT A FT14112VCB74):

T24

```
 ------------------------------------------------------------------------------
 16 DEC 2014 21:25:52  USER (22 APR) VLADIMIR.K        [15803,IPAGE 1   >>>3>>>
 ACTION                                              YOU HAVEN'T SEEN ALL PAGES
 AWAITING PAGE INSTRUCTIONS
```

Yes, we have to go through all pages to be able to authorise; after successful authorisation new values appear in the field STMT.NOS:

T24

```
 195. 1 STMT.NOS....... 171521580377286.00
 195. 2 STMT.NOS....... 1-2
```

This is the link to accounting entries.  Besides that, WORKING.BALANCE for both accounts changed:

jsh

```
LIST FBNK.ACCOUNT '14378' '66419' WORKING.BALANCE
```

Output

```
@ID.............    WORKING.BALANCE....

              14378               23213.47
              66419                     30
```

There’s also information about starting balance and the last transaction in ACCOUNT record:

T24

```
     ACCOUNT.NUMBER.... 14378               MICK ROSS
 ------------------------------------------------------------------------------
...
  23 OPEN.ACTUAL.BAL... 23,243.47
...
  37 DATE.LAST.DR.CUST. 22 APR 2014
  38 AMNT.LAST.DR.CUST. -30.00
  39 TRAN.LAST.DR.CUST. 213                Transfer Out
```

Enrichment for TRAN.LAST.DR.CUST was taken from application TRANSACTION:

T24

```
 Model Bank R14          TRANSACTION SEE

     TRANSACTION.CODE.. 213
 ------------------------------------------------------------------------------
   1. 1 GB NARRATIVE... Transfer Out
  10 TURNOVER.CHARGE... NO
  11 SWIFT.NARRATIVE... NTRF
  13 INITIATION........ CUSTOMER
  14 SHORT.DESC........ Transfer
  15 STMT.NARR......... Transfer
  16 STMT.NARR.REF..... THEIR
  18 NARR.TYPE......... FTOUT               FTOUT
  26 DISPO.EXEMPT...... YES
  35 CURR.NO........... 3
  36. 1 INPUTTER....... 1_201214m
  37. 1 DATE.TIME...... 28 MAR 14 13:43
  38 AUTHORISER........ 68605_AUTHORISER_OFS_MB.OFS.AUTH
  39 CO.CODE........... GB-001-0001         Model Bank R14
  40 DEPT.CODE......... 1                   Implementation

 ------------------------------------------------------------------------------
 16 DEC 2014 21:34:33  USER (22 APR) INPUTTER          [17785,IPAGE 1
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

...and we can double-check it in SS record:

T24

```
 Model Bank R14          STANDARD SELECTION FIELDS SEE

     FILE.NAME......... ACCOUNT
 ------------------------------------------------------------------------------
...
   1.55 SYS.FIELD.NAME. TRAN.LAST.DR.CUST
   2.55 SYS.TYPE....... D
   3.55. 1 SYS.FIELD.NO 39
   4.55. 1 SYS.VAL.PROG IN2&&EXTERN
   6.55 SYS.DISPLAY.FMT 3R
   7.55 SYS.ALT.INDEX.. N
  10.55 SYS.SINGLE.MULT S
  11.55 SYS.LANG.FIELD. N
  12.55 SYS.GENERATED.. Y
  14.55. 1 SYS.REL.FILE TRANSACTION         Transaction code definition for Tr|
...
```

Also, monthly record in ACCT.ACTIVITY (L-type application) was updated for both accounts:

T24

```
Model Bank R14          Account Activity SEE
                                          CHF  Current Account
    ACCT.NO.YEAR.MONTH 14378 - 2014-04    MICK ROSS
--------------------------------------------------------------------------------
  1. 1 DAY.NO......... 22
  3. 1 TURNOVER.DEBIT. -30.00
  4. 1 BALANCE........ 23,213.47
  5. 1 TRANSACT.CODE.. 213                Transfer Out
  6. 1 NO.OF.TRANSACT. 1
  7. 1 TRANSACT.AMT... -30.00
 15. 1 BK.DAY.NO...... 22
 16. 1 BK.BALANCE..... 23,213.47
 18. 1 BK.DEBIT.MVMT.. -30.00









--------------------------------------------------------------------------------
16 DEC 2014 21:36:41  USER (22 APR) INPUTTER          [17785,IPAGE 1
ACTION
AWAITING PAGE INSTRUCTIONS
```

T24

```
Model Bank R14          Account Activity SEE
                                          CHF  Current Account
    ACCT.NO.YEAR.MONTH 66419 - 2014-04    Acorn Engineering Company
--------------------------------------------------------------------------------
  1. 1 DAY.NO......... 22
  2. 1 TURNOVER.CREDIT 30.00
  4. 1 BALANCE........ 30.00
  5. 1 TRANSACT.CODE.. 258                Transfer In
  6. 1 NO.OF.TRANSACT. 1
  7. 1 TRANSACT.AMT... 30.00
 15. 1 BK.DAY.NO...... 22
 16. 1 BK.BALANCE..... 30.00
 17. 1 BK.CREDIT.MVMT. 30.00









--------------------------------------------------------------------------------
16 DEC 2014 21:38:23  USER (22 APR) INPUTTER          [17785,IPAGE 1
ACTION
AWAITING PAGE INSTRUCTIONS
```

## 1.66 Accounting entries

In the previous chapter we entered FT deal that upon authorisation created accounting entries:

T24 - FT record

```
195. 1 STMT.NOS....... 171521580377286.00
195. 2 STMT.NOS....... 1-2
```

Appending 0001 and 0002 for @IDs to the first value of STMT.NOS we can find them in STMT.ENTRY application (“L” type again):

T24

```
Model Bank R14          STMT.ENTRY SEE

    STMT.ENTRY.ID..... 171521580377286.000001
--------------------------------------------------------------------------------
  1 ACCOUNT.NUMBER.... 14378               Mick Ross
  2 COMPANY.CODE...... GB-001-0001         Model Bank R14
  3 AMOUNT.LCY........ -33.93
  4 TRANSACTION.CODE.. 213                 Transfer Out
  8 CUSTOMER.ID....... 100285              Mick Ross
  9 ACCOUNT.OFFICER... 74                  PWM Relationship Manager
 10 PRODUCT.CATEGORY.. 1-001               Current Account
 11 VALUE.DATE........ 22 APR 2014
 12 CURRENCY.......... CHF                 Swiss Franc
 13 AMOUNT.FCY........ -30.00
 14 EXCHANGE.RATE..... 0.884173298
 16 POSITION.TYPE..... TR                  TRADING POSITION
 17 OUR.REF FT14112VCB74
 19 EXPOSURE.DATE..... 22 APR 2014
 20 CURRENCY.MARKET... 1                   Currency Market
 22 DEPARTMENT.CODE... 1                   Implementation
 23 TRANS.R FT14112VCB74
 24 SYSTEM.ID......... FT                  FUNDS TRANSFER
 25 BOOKING.DATE...... 22 APR 2014
 26. 1 STMT.NO........ 171521580377286.00
 26. 2 STMT.NO........ 1-2
 29 CURR.NO........... 1
 30. 1 INPUTTER....... 29528_INPUTTER
 31. 1 DATE.TIME...... 16 DEC 14 21:28
 32 AUTHORISER........ 15803_VLADIMIR.K
 36 CRF.TYPE.......... CREDIT
 39 CONSOL. AC.1.TR.CHF.1001.1001.US.....1000.....GB0010001
 91 NET.PARAM......... CLDEFAULT
100 PROCESSING.DATE... 22 APR 2014
104 ORIG.CCY.MARKET... 1
--------------------------------------------------------------------------------
```

T24

```
Model Bank R14          STMT.ENTRY SEE

    STMT.ENTRY.ID..... 171521580377286.000002
------------------------------------------------------------------------------
  1 ACCOUNT.NUMBER.... 66419               Acorn Engineering Company
  2 COMPANY.CODE...... GB-001-0001         Model Bank R14
  3 AMOUNT.LCY........ 33.93
  4 TRANSACTION.CODE.. 258                 Transfer In
  8 CUSTOMER.ID....... 128053              Acorn Engineering Company
  9 ACCOUNT.OFFICER... 74                  PWM Relationship Manager
 10 PRODUCT.CATEGORY.. 1-001               Current Account
 11 VALUE.DATE........ 22 APR 2014
 12 CURRENCY.......... CHF                 Swiss Franc
 13 AMOUNT.FCY........ 30.00
 14 EXCHANGE.RATE..... 0.884173298
 16 POSITION.TYPE..... TR                  TRADING POSITION
 17 OUR.REF FT14112VCB74
 19 EXPOSURE.DATE..... 22 APR 2014
 20 CURRENCY.MARKET... 1                   Currency Market
 22 DEPARTMENT.CODE... 1                   Implementation
 23 TRANS.R FT14112VCB74
 24 SYSTEM.ID......... FT                  FUNDS TRANSFER
 25 BOOKING.DATE...... 22 APR 2014
 26. 1 STMT.NO........ 171521580377286.00
 26. 2 STMT.NO........ 1-2
 29 CURR.NO........... 1
 30. 1 INPUTTER....... 29528_INPUTTER
 31. 1 DATE.TIME...... 16 DEC 14 21:28
 32 AUTHORISER........ 15803_VLADIMIR.K
 36 CRF.TYPE.......... CREDIT
 39 CONSOL. AC.1.TR.CHF.1001.2001.US.....2710.....GB0010001
 91 NET.PARAM......... CLDEFAULT
100 PROCESSING.DATE... 22 APR 2014
104 ORIG.CCY.MARKET... 1
------------------------------------------------------------------------------
```

STMT.ENTRY is usually very big; though we have transaction reference there, it would be very long to select this table - its @IDs are generated based on current time, terminal number etc and do not possess any business meaning.  We can use a concat file ACCT.ENT.TODAY to see what was generated for particular account during the current work day:

T24

```
Model Bank R14          Account Entries for Today SEE

    ACCOUNT.NO........ 66419               Acorn Engineering Company
------------------------------------------------------------------------------
  1 ENTRY.NO.......... 171521580377286.000002
```

If there’s more than one entry, it will be shown as another field rather than a value:

T24

```
Model Bank R14          Account Entries for Today SEE

    ACCOUNT.NO........ 34517              FIDELITY EQUITY-CONTROL ACC
--------------------------------------------------------------------------------
  1 ENTRY.NO.......... 169424542232486.000001
  2 ENTRY.NO.......... 169424542232500.030001
  3 ENTRY.NO.......... 169424542232506.030001
  4 ENTRY.NO.......... 169424542232514.000001
  5 ENTRY.NO.......... 169424542232520.020001
  6 ENTRY.NO.......... 169424542232526.030001
  7 ENTRY.NO.......... 169424542232533.030001
  8 ENTRY.NO.......... 169424542232540.010001
  9 ENTRY.NO.......... 169424542232546.020001
 10 ENTRY.NO.......... 169424542232552.030001









--------------------------------------------------------------------------------
16 DEC 2014 21:54:24  USER (22 APR) INPUTTER          [17469,IPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

This table is saved to ACCT.ENT.LWORK.DAY in the end of day so yesterday’s entries can also be found easily.

The table structure is not usual and it’s a bit tricky to find some particular ENTRY.NO. Search for ENTRY.NO #5 at the screen above:

jsh

```
LIST FBNK.ACCT.ENT.TODAY WITH ENTRY.NO EQ '169424542232520.020001'
```

No records found...  If we take the #1 then it will be found:

jsh

```
LIST FBNK.ACCT.ENT.TODAY WITH ENTRY.NO EQ '169424542232486.000001'
```

Output

```
@ID........ 34517
ACCOUNT.NO. 34517
ENTRY.NO... 169424542232486.000001
```

ENTRY.NO is considered as field #1 and the entry 169424542232520.020001 that wasn't found is located at the field #5:

jsh

```
LIST FBNK.ACCT.ENT.TODAY WITH *A5 EQ '169424542232520.020001'
```

It gives us the same output as above.  Since we don't know the position of what we search, the task can be solved by either matching the whole record contents...

jsh

```
LIST FBNK.ACCT.ENT.TODAY WITH EVAL "@RECORD" LIKE '...169424542232520.020001...'
```

...or searching the whole file:

jsh

```
SEARCH FBNK.ACCT.ENT.TODAY
String :169424542232520.020001
String :
Record Keys : *

 1 Records selected

>
```

In case of income/expense operation (like charge etc) the entry will be placed not to STMT.ENTRY but to CATEG.ENTRY table.  Example:

jsh

```
LIST FBNK.CATEG.ENTRY SAMPLE 1
```

Output

```
@ID................ 169372212577430.130001
CATEG.ENTRY.ID..... 169372212577430.130001
OUR.REFERENCE......
PL.CATEGORY........ 50001
CUSTOMER.ID........
CURRENCY...........
PRODUCT.CATEGORY...
ACCOUNT.NUMBER.....
COMPANY.CODE....... GB0010001
AMOUNT.LCY......... -5.07
TRANSACTION.CODE...
...
```

Here we have not account number but PL (profit&loss) CATEGORY 50001:

T24

```
Model Bank R14          Category SEE

    CATEGORY.CODE..... 50-001
------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. Interest Expense (1)
  1. 2 FR DESCRIPTION. Intérêts payés
  2. 1 GB SHORT.NAME.. Interest Exp(1)
  3 SYSTEM.IND........ PL
 20 CURR.NO........... 4
 21. 1 INPUTTER....... 36247_INPUTTER__OFS_MB.LANG
 22. 1 DATE.TIME...... 02 JUN 14 15:52
 23 AUTHORISER........ 36247_INPUTTER_OFS_MB.LANG
 24 CO.CODE........... GB-001-0001         Model Bank R14
 25 DEPT.CODE......... 1                   Implementation









------------------------------------------------------------------------------
16 DEC 2014 21:48:46  USER (22 APR) INPUTTER          [1501,INPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

In T24 the concept of “one debit-one credit” is not mandatory - entries are one-sided.

All amounts for the day summed up should show the same debit and credit totals in local currency (the third application - RE.CONSOL.SPEC.ENTRY - also takes part in the balancing).

It’s possible to write a hook subroutine that is triggered every time when accounting entry is raised online - see field ACCOUNTING.SUBRTN in ACCOUNT.PARAMETER application.

*(At COB - close of business - accounting entries are also raised - e.g.  when doing interest capitalization but at that stage the ACCOUNTING.SUBRTN isn’t triggered.)*

In STMT.ENTRY 171521580377286.000001 there was a field CONSOL.KEY with the value “AC.1.TR.CHF.1001.1001.US.....1000.....GB0010001”.  This is the reflection of the next step of consolidation of accounting entries.

See the corresponding record in application CONSOLIDATE.ASST.LIAB:

T24

```
Model Bank R14          Consolidate Asset Liabilities SEE

    CONSOL.KEY........ AC.1.TR.CHF.1001.1001.US.....1000.....GB0010001
------------------------------------------------------------------------------
  1 APPLIC.ID......... AC
  2 CURRENCY.MARKET... 1                   Currency Market
  3 POSITION.TYPE..... TR                  TRADING POSITION
  4 CURRENCY.......... CHF                 Swiss Franc
  5 CATEGORY.......... 1-001               Current Account
  6 SECTOR............ 1001                Individual
  7 RESIDENCE......... US                  United States of America
 12 INDUSTRY.......... 1000                Private Person (Name)
 17 DATE.LAST.UPDATE.. 17 APR 2014
 18. 1 TYPE........... 50000
 19. 1 BALANCE........ 56.04
 21. 1 CREDIT.MOVEMENT 17.51
 22. 1 LOCAL.BALANCE.. 63.37
 24. 1 LOCAL.CREDT.MVE 19.80
 18. 2 TYPE........... CREDIT
 19. 2 BALANCE........ 23,243.47
------------------------------------------------------------------------------
03 FEB 2015 18:08:38  USER (22 APR) VLADIMIR.K        [12583,IPAGE 1   >>>2>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

Consolidation parameters are visible in the fields 4 to 12 and are set during initial T24 setup in application CONSOLIDATE.COND.

Profits and losses are consolidated into CONSOLIDATE.PRFT.LOSS.

During Close of Business - COB - accounting (or “GL” - General Ledger) reports are created also, but this is almost totally a business issue and therefore isn’t described in more detail here - it will be enough just to mention names of setup applications:  RE.STAT.REP.LINE, RE.STAT.LINE.CONT, RE.STAT.RANGE, RE.STAT.REPORT.HEAD and others; there are also some tables that a technical person might use - for example, for a local report:  RE.STAT.REQUEST, RE.STAT.LINE.BAL, RE.STAT.MISMATCH, EB.SYSTEM.SUMMARY.

## 1.67  tSM and tSA agents

tSM (Temenos Service Manager) is T24 core jBC program that is responsible for activating and managing so-called “tSA agents” in order to proceed with both online and end-of-day tasks.  Setup records for both tSM and tSA agents are stored in application TSA.SERVICE.

Record for tSM:

T24

```
Model Bank R14          TSA.SERVICE INPUT

    SERVICE........... TSM
--------------------------------------------------------------------------------
  1. 1 DESCRIPTION.... TSM Record for running upgrade as service
  2. 1 SERVER.NAME....
  3. 1 WORK.PROFILE... TSM                 WORK LOAD PROFILE FOR TSM
  4. 1 SERVER.STATUS..
  5 USER.............. INPUTTER            INPUTTER
  6 SERVICE.CONTROL... START
  7 REVIEW.TIME.......
  8 TIME.OUT..........
  9. 1 ATTRIBUTE.TYPE.
 10. 1 ATTRIBUTE.VALUE
 11 FREQUENCY.........
 12 RESERVED.8........
 13 RESERVED.7........
 14. 1 LOCAL.REF......
 15. 1 DATE........... 22 APR 2014
 16. 1 STARTED........ 22/05/2014 15:00:33
--------------------------------------------------------------------------------
17 DEC 2014 00:43:39  USER (22 APR) INPUTTER          [14282,IPAGE 1   >>>4>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

*(tSA agents have nothing in common with jbase agents except the name.)*

To start tSM set SERVICE.CONTROL to START, commit the record and then enter command START.TSM from jsh prompt.

Output

```
START.TSM
Phantom process started on process id 3232
 [3232] Done : tSA 1
```

WHERE output

```
...
 3      VT100      r14          3232     jsh -Jz -c tSA 1
                                         tSA 1
                                         tSM
...
```

Then tSA agents that have SERVICE.CONTROL set to START or AUTO will be activated by tSM. Those who have “START” will be executed and then stopped (and field SERVICE.CONTROL will be set to STOP). Those who are set to AUTO will be executed repeatedly until tSM is stopped.

To stop tSM set its SERVICE.CONTROL field to STOP and wait for some time.  tSM will stop by itself.

This can be done in one step at T24 “AWAITING APPLICATION” prompt:

T24

```
TSA.SERVICE, I TSM 6 S
```

...then F5, that’s all.

tSM and tSA agents log their activities to &COMO& records with names starting from “tSA” followed by agent number and date/time of starting with underscore as de-limiter, e.g.:

jsh

```
CT &COMO& tSA_1_20141217_00-48-12
```

Output

```
    tSA_1_20141217_00-48-12
001 COMO tSA_1_20141217_00-48-12 established 00:48:13  17 DEC 2014
002 <como>
003 <agent>1</agent>
004 <processid>3232</processid>
005 <portno>3</portno>
006 Agent 1 started 17 DEC 14 00-48-12
007 Agent's Process id 3232
008 <servername>r14vm</servername>
009 Running on server r14vm PortNumber 3
010 <service name = TSM>
011 tSM
```

tSM runs in one session only; if START.TSM is accidentally executed again, it won’t be duplicated:

COMO contents

```
001 COMO tSA_10_20141217_00-54-59 established 00:54:59  17 DEC 2014
002 <como>
003 <agent>10</agent>
004 <processid>3396</processid>
005 <portno>4</portno>
006 Agent 10 started 17 DEC 14 00-54-59
007 Agent's Process id 3396
008 <servername>r14vm</servername>
009 Running on server r14vm PortNumber 4
010 <service name = TSM>
011 tSM
012 tSM already running on Server r14vm                                  <=====
013 </service>
014 Agent stopped                                                       <=====
015 </como>
016 COMO completed 00:55:00  17 DEC 2014
```

Number of sessions for tSA agents is set in TSA.SERVICE via field WORK.PROFILE.

T24

```
Model Bank R14          TSA.SERVICE, INPUT

    SERVICE........... BNK/SWIFT.OUT
------------------------------------------------------------------------------
  1. 1 DESCRIPTION.... DE FORMATTING SERVICE
  2. 1 SERVER.NAME....
  3. 1 WORK.PROFILE... DE.FORMATTING.SERVICE USED FOR DE FORMATTING SERVICE
  4. 1 SERVER.STATUS..
  5 USER.............. INPUTTER            INPUTTER
  6 SERVICE.CONTROL... STOP
  7 REVIEW.TIME....... 10
  8 TIME.OUT..........
  9. 1 ATTRIBUTE.TYPE.
 10. 1 ATTRIBUTE.VALUE
 11 FREQUENCY.........
 12 RESERVED.8........
 13 RESERVED.7........
 14. 1 LOCAL.REF......
 15. 1 DATE........... 22 APR 2014
 16. 1 STARTED........ 06/06/2014 08:25:08
------------------------------------------------------------------------------
17 DEC 2014 01:07:45  USER (22 APR) INPUTTER          [28738,IPAGE 1   >>>3>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Even if there is something like “ONE” or “TWO” - it’s not actual numbers but a link to TSA.WORKLOAD.PROFILE:

T24

```
Model Bank R14          TSA.WORKLOAD.PROFILE INPUT

    WORKLOAD.PROFILE.. DE.FORMATTING.SERVICE
------------------------------------------------------------------------------
  1 DESCRIPTION....... USED FOR DE FORMATTING SERVICE
  2. 1 TIME...........
  3. 1 AGENTS.REQUIRED 2
```

*(Sometimes in “ONE” record there’s 5 AGENTS.REQUIRED...)*

tSA agents also have a corresponding entry in BATCH application:

T24

```
Model Bank R14          BATCH ENTRY SEE

    BATCH PROCESS..... BNK/SWIFT.OUT
--------------------------------------------------------------------------------
  3 PROCESS.STATUS.... 0                   READY
  4 BATCH.ENVIRONMENT. F                   FOREGROUND
  6. 1 JOB.NAME....... DE.OUTWARD          JOB FOR FORMATTING SERVICE
  8. 1 FREQUENCY...... D                   DAILY
  9. 1 NEXT.RUN.DATE.. 19 MAR 2014
 12. 1 JOB.STATUS..... 0                   READY
 28 CURR.NO........... 1
 29. 1 INPUTTER....... SY_COMP.IN.BATCH
 30. 1 DATE.TIME...... 10 APR 14 17:44
 31 AUTHORISER........ 2_AUTHORISER_OFS_MB.OFS.AUTH
 32 CO.CODE........... GB-001-0001         Model Bank R14
 33 DEPT.CODE......... 1                   Implementation




--------------------------------------------------------------------------------
03 FEB 2015 18:36:54  USER (22 APR) VLADIMIR.K        [17093,IPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Field JOB.NAME refers to the entry in PGM.FILE. We already saw PGM.FILE records for applications (H, U, L, T, W, D) and for subroutines (S); this one has type B reserved for online services and COB jobs:

T24

```
Model Bank R14          PROGRAM FILE SEE

    PROGRAM            DE.OUTWARD
--------------------------------------------------------------------------------
  1 TYPE.............. B
  2. 1 GB SCREEN.TITLE JOB FOR FORMATTING SERVICE
  4. 1 BATCH.JOB...... @BATCH.JOB.CONTROL
  5 PRODUCT........... DE
  7. 1 DESCRIPTION.... Formats the outward messages with
  7. 2 DESCRIPTION.... respect to the carrier and also
  7. 3 DESCRIPTION.... sends the formatted message to the
  7. 4 DESCRIPTION.... interface if any
  9 ACTIVATION.FILE... YES
 26 CURR.NO........... 3
 27. 1 INPUTTER....... 1_R09.000m
 28. 1 DATE.TIME...... 03 APR 09 00:54
 29 AUTHORISER........ 1_R09.000m
 30 CO.CODE........... GB-001-0001         Model Bank R14
 31 DEPT.CODE......... 1                   Implementation

--------------------------------------------------------------------------------
```

## 1.68 COB

COB is the process of finishing the work day and starting a new one. Corresponding tSA agent is to be started to initialize the end-of-day activities:

T24

```
Model Bank R14          TSA.SERVICE, INPUT

    SERVICE........... COB
--------------------------------------------------------------------------------
  1. 1 DESCRIPTION.... FOR COB
  2. 1 SERVER.NAME....
  3. 1 WORK.PROFILE... TWO                 FOR COB
  4. 1 SERVER.STATUS..
  5 USER.............. INPUTTER            INPUTTER
  6 SERVICE.CONTROL... STOP
  7 REVIEW.TIME.......
  8 TIME.OUT..........
  9. 1 ATTRIBUTE.TYPE.
 10. 1 ATTRIBUTE.VALUE
 11 FREQUENCY.........
 12 RESERVED.8........
 13 RESERVED.7........
 14. 1 LOCAL.REF......
 15. 1 DATE........... 17 APR 2014
 16. 1 STARTED........ 16/05/2014 05:22:33
--------------------------------------------------------------------------------
17 DEC 2014 01:10:17  USER (22 APR) INPUTTER         [28738,IPAGE 1   >>>5>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

SERVICE.CONTROL should be set to START and not to AUTO since we normally want it to be executed only once.

Before starting COB check EB.EOD.ERROR application for errors raised during previous COB (screenshot is taken from R11 Model Bank):

T24

```
    EB.EOD.ERROR.ID... GB0010001.20110414
--------------------------------------------------------------------------------
  1. 1 TIME.DATE...... 10:06:22 06 MAY 2011
  2. 1. 1 DESCRIPTION. RECORD NOT FOUND
  3. 1 APPLICATION.ID. SC                  SECURITIES
  4. 1 ROUTINE........ (SC.OL.VAL.SEC)
  5. 1. 1 RECORD.KEY.. CURRENT.PORTFOLIO
  6. 1. 1 DETAIL.KEY.. 158320665436382.00
  7. 1 DATE.RESOLVED.. _
  8. 1 FIX.REQUIRED... NO
  9 RESERVED.12.......
 10 RESERVED.11.......
 11 RESERVED.10.......
 12 RESERVED.9........
 13 RESERVED.8........
 14 RESERVED.7........
 15 RESERVED.6........
 16 RESERVED.5........
```

Put the current date to all values (may be quite a few) of DATE.RESOLVED field and commit as soon as the all problems described in this record are resolved, otherwise COB won't start.

More detail about errors can be found in application EB.EOD.ERROR.DETAIL with @ID mentioned in the field DETAIL.KEY of EB.EOD.ERROR record:

T24

```
Model Bank              EB.EOD.ERROR.DETAIL SEE

    DETAIL.ID......... 158320665436382.00
------------------------------------------------------------------------------
  1 DATE.AND.TIME..... 10:06:22 06 MAY 2011
  2 APPLIC.ID......... SC                  SECURITIES
  3 ROUTINE........... SC.OL.VAL.SEC - START.UPDATE
  4 MODULE............ SC
 13 SYSTEM.DATE....... 14 APR 2011
```

COB is executed by so-called "batches":

jsh

```
LIST F.BATCH BATCH.STAGE
```

Output

```
@ID.............................................    BATCH.STAGE

MF1/RC.CYCLER.ONLINE.SERVICE
EU1/SC.SETTLEMENT.SERVICE
BR1/RE.BASE.CCY.CRF.SERVICE
SG1/BAL.CHK.LISTENER.SERVICE
MF1/LD.EOD.REPORTS                                 A101
MF2/SL.EOD.REPORTS                                 A101
EU1/DM.END.OF.DAY3                                 A910
```

Part of @ID before slash - company @ID. BATCH.STAGE is empty for online services (described in the previous chapter), COB batches have this field populated.

COB executes batches in the following sequence of stages:  A (application end of day), S (system end of day), R (reporting), D (start of new day), O (online).  Batches with the same stage can be executed simultaneously (like MF1/LD.EOD.REPORTS and MF2/SL.EOD.REPORTS at the screenshot above).

The first thing COB does is:

COMO contents

```
_0__4_06 NOV 2014_08:27:02_SSELECT F.BATCH BY BATCH.STAGE Selected=1421 time=2secs
```

Current work day is changed immediately and it’s reflected in DATES records, so - if T24 module NS (“Non-stop”) module is installed - users can input transactions to the next work day:

T24

```
Model Bank R14           DATES - Default List

   ID               TODAY        VERSION  LAST.WORKING.DAY  PERIOD.END  FUNCT.
------------------------------------------------------------------------------
 1  EU0010001        23 APR 2014        1       22 APR 2014     20140423
 2  EU0010001-COB    22 APR 2014        1       17 APR 2014     20140422
 3  GB0010001        23 APR 2014        1       22 APR 2014     20140423
 4  GB0010001-COB    22 APR 2014        1       17 APR 2014     20140422
 5  GB0010002        23 APR 2014        1       22 APR 2014     20140423
 6  GB0010002-COB    22 APR 2014        1       17 APR 2014     20140422
 7  GB0010003        23 APR 2014        1       22 APR 2014     20140423
 8  GB0010003-COB    22 APR 2014        1       17 APR 2014     20140422
 9  GB0010004        23 APR 2014        1       22 APR 2014     20140423
10  GB0010004-COB    22 APR 2014        1       17 APR 2014     20140422
11  GB0010005        23 APR 2014        1       22 APR 2014     20140423
12  GB0010005-COB    22 APR 2014        1       17 APR 2014     20140422
13  SG0010001        23 APR 2014        1       22 APR 2014     20140423
14  SG0010001-COB    22 APR 2014        1       17 APR 2014     20140422


------------------------------------------------------------------------------
17 DEC 2014 02:37:26  USER (23 APR) INPUTTER          [8159,INPAGE   1 >>>1>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

*(If T24 module GP - “Global Processing” - is installed, it’s possible to run COB not for all the bank but for some particular branch or group of branches.)*

Each batch contains one or several jobs:

T24

```
Model Bank R14          BATCH ENTRY SEE

    BATCH PROCESS..... BNK/FT.END.OF.DAY
------------------------------------------------------------------------------
  1 BATCH.STAGE....... A100               APPLICATION
  3 PROCESS.STATUS.... 0                  READY
  4 BATCH.ENVIRONMENT. F                  FOREGROUND
  6. 1 JOB.NAME....... FT.END.OF.DAY.2    FT.END.OF.DAY.2                    <===
  8. 1 FREQUENCY...... D                  DAILY
  9. 1 NEXT.RUN.DATE.. 23 JUL 1998
 12. 1 JOB.STATUS..... 0                  READY
 13. 1 LAST.RUN.DATE.. 17 APR 2014
  6. 2 JOB.NAME....... FT.PROCESS.BULK.PAYMENT.PRE FT PROCESS BULK PAYMENT PRE  <===
  8. 2 FREQUENCY...... D                  DAILY
 12. 2 JOB.STATUS..... 0                  READY
 13. 2 LAST.RUN.DATE.. 17 APR 2014
  6. 3 JOB.NAME....... FT.PROCESS.BULK.PAYMENT FT PROCESS BULK PAYMENT          <===
  7. 3. 1 VERIFICATION FT.PROCESS.BULK.PAYMENT.PRE FT PROCESS BULK PAYMENT PRE
...
```

VERIFICATION sets the jobs dependencies, e.g.  in the example above the job #3 - FT.PROCESS.BULK.PAYMENT - depends on successful completion of the job #2.

FREQUENCY=D means to run daily (NEXT.RUN.DATE was used for it someday but now it's not touched as you can see), other mostly used frequencies are W (weekly), M (monthly), A (adhoc).  To set adhoc job for running its NEXT.RUN.DATE has to be set to current work day (or to the next work day for start-of-day batch).

Each tSA agent running COB also logs its output to COMO, e.g.:

COMO contents

```
_BNK/PD.END.OF.DAY_PD.END.OF.DAY_16_01 OCT 2014_10:38:05_Obtained the Locking with
 BNK/PD.END.OF.DAY_PD.END.OF.DAY-1 and F.JOB.LIST.2
_BNK/PD.END.OF.DAY_PD.END.OF.DAY_16_01 OCT 2014_10:38:05_Using list file F.JOB.LIST.2
_BNK/PD.END.OF.DAY_PD.END.OF.DAY_16_01 OCT 2014_10:38:05_Control list processing 10 59
```

Same as it was for tSA services, each job has a corresponding entry in PGM.FILE (type B):

T24

```
Model Bank R14          PROGRAM FILE, SEE

    PROGRAM            FT.END.OF.DAY.2
------------------------------------------------------------------------------
  1 TYPE.............. B
  2. 1 GB SCREEN.TITLE FT.END.OF.DAY.2
  3 ADDITIONAL.INFO... R
  4. 1 BATCH.JOB...... @BATCH.JOB.CONTROL
  5 PRODUCT........... FT
  7. 1 DESCRIPTION.... Job to move live FT records to
  7. 2 DESCRIPTION....  History file
 26 CURR.NO........... 3
 27. 1 INPUTTER....... 1_200810m
 27. 2 INPUTTER....... 49_CONV.PGM.FILE.R09
 28. 1 DATE.TIME...... 16 OCT 08 15:30
 29 AUTHORISER........ 1_200810m
 30 CO.CODE........... GB-001-0001         Model Bank R14
 31 DEPT.CODE......... 1                   Implementation


------------------------------------------------------------------------------
17 DEC 2014 02:00:43  USER (22 APR) INPUTTER          [6517,INPAGE 1
ACTION
AWAITING PAGE INSTRUCTIONS
```

BATCH.JOB......  @BATCH.JOB.CONTROL means that it's so-called "multi-threaded" job (though the correct name would have to be "multi-session").  This job can be executed by more than one tSA agent.

Standard multi-threaded job consists of 3 parts:

• SELECT routine.  It's being run only once; it selects records for processing and puts them to F.JOB.LIST.n table, where n is a number assigned by tSA agent that proceeded with the select.

• Record processing routine.  It obtaines record @ID for procesing from F.JOB.LIST.n, locks F.JOB.LIST.n record and deletes it when the processing is finished.  When there’s no more records in F.JOB.LIST.n, next job is ready for execution.

• LOAD routine.  Called from record processing routine to load COMMON areas with global variables like file handles etc.

Distribution of data in F.JOB.LIST isn’t 1:1, some records might contain one @ID to process, some - many of them:

jsh

```
LIST F.JOB.LIST.1
```

Sample output

```
    1
001 1

    10
001 100004

    100
001 100088

    1000
001 97989
002 9799
003 97990
004 97992
005 97993
006 97994
007 97995
008 97996
...
```

To count all of them use DCOUNT function inside jQL statement EVAL:

jsh

```
LIST F.JOB.LIST.1 EVAL "DCOUNT(@RECORD,@FM)" AS CNT TOTAL CNT DET-SUPP
```

Sample output

```
@ID.......    CNT.......    CNT.......

***                         163820

 1020 Records Listed
```

None of F.JOB.LIST.n tables is T24 application:

T24

```
-- LAST SIGN.ON, DATE: 03 FEB 2015     TIME: 18:36        ATTEMPTS: 0 --------
03 FEB 2015 18:54:51  USER (22 APR) VLADIMIR.K        [24163,IN]
ACTION F.JOB.LIST.1                                        APPLICATION MISSING
AWAITING APPLICATION
```

They are work files that don’t have PGM.FILE entry.

In case of a fatal error the record in EB.EOD.ERROR is created (or updated) with error details and record @ID is removed from F.JOB.LIST.

Fatal error messages can also be found in COMO:

COMO contents

```
_BNK/PD.END.OF.DAY_PD.END.OF.DAY.2_4_06 NOV 2014_08:27:03_Fatal error in CDD error
   DATE WITHOUT 8 CHARACTERS
```

or:

COMO contents

```
<job name = PD.EOD.CONTRACT>
_BNK/PD.END.OF.DAY_PD.EOD.CONTRACT_26_17 SEP 2014_16:50:03_Standard multi-thread job
_BNK/PD.END.OF.DAY_PD.EOD.CONTRACT_26_17 SEP 2014_16:50:03_Calling load routine
_BNK/PD.END.OF.DAY_PD.EOD.CONTRACT_26_17 SEP 2014_16:50:03_Obtained the Locking with
    BNK/PD.END.OF.DAY-PD.EOD.CONTRACT-5 and F.JOB.LIST.6
_BNK/PD.END.OF.DAY_PD.EOD.CONTRACT_26_17 SEP 2014_16:50:04_Using list file
    F.JOB.LIST.6
_BNK/PD.END.OF.DAY_PD.EOD.CONTRACT_26_17 SEP 2014_16:50:04_Control list PROCESS.PDS
    processing 225 4455 PURGE.EXCEPTION
_BNK/PD.END.OF.DAY_PD.EOD.CONTRACT_26_17 SEP 2014_16:50:04_SELECT F.JOB.LIST.6 SAMPLE
   100000 Selected=1 time=0secs
 ** Error [ NOT_FILE_VAR ] **
Variable is not an opened file descriptor , Line   144 , Source F.RELEASE
 ** Error [ EXIT_TRANS_ABORT ] **
TRANSACTION ABORT forced as transaction open when program terminated
```

If COB stopped (field SERVICE.CONTROL of COB record in TSA.SERVICE is set to STOP) but hadn’t fully finished, it could be that some job totally failed.  To find it see in which F.BATCH record the field PROCESS.STATUS has the value 1:

jsh

```
LIST F.BATCH WITH BATCH.STAGE NE '' AND PROCESS.STATUS EQ 1
```

Sample output

```
BATCH.PROCESS...... BNK/LD.END.OF.DAY
BATCH.STAGE........ A100
DEFAULT.PRINTER....
PROCESS.STATUS..... 1
BATCH.ENVIRONMENT.. F
DEPARTMENT.CODE....
JOB.NAME........... LD.PD.COMMT.UPDATE   LD.EOD.9   LD.PERIODIC.ADJUSTMENTS
                    LD.END.OF.DAY.4   LD.PROCESS.PROVISION   LD.END.OF.DAY.8
                    LD.END.OF.DAY.3   LD.EOD.ISSUE.REIMB.ACCRUAL
                    LD.END.OF.DAY.2   LD.POST.END.OF.DAY.2   LD.EOD.5
VERIFICATION....... LD.EOD.9   LD.EOD.9   LD.END.OF.DAY.4   LD.END.OF.DAY.4
                    LD.END.OF.DAY.4   LD.END.OF.DAY.3   LD.END.OF.DAY.3
                    LD.END.OF.DAY.2   LD.POST.END.OF.DAY.2
FREQUENCY.......... D   D   D   D   D   D   D   D   D   D   D
NEXT.RUN.DATE...... 19980723      19980723         19980723   19980723
                    19980723   19980723   19980723
PRINTER.NAME.......
JOB.STATUS......... 2   2   1   0   0   0   0   0   0   0   0   0
```

JOB.STATUS=1 shows us which job is “running”, i.e.  hadn’t finished.  0 means “didn’t run yet”, 2 - finished (after COB completion all 2’s are replaced to 0’s); 3 means “fatal error”.

In some cases erroneous job can be skipped (e.g.  a local one) by correcting F.BATCH record in JED and clearing of corresponding F.JOB.LIST.

*(Batches with the same stage can be run simultaneously so there can be several populated F.JOB.LIST files.)*

Batch COB.INITIALISE for each company (e.g.  BNK/COB.INITIALISE, EU1/COB.INITIALISE etc) contains job COB.EXECUTE.API that triggers the local routine if it’s mentioned in SPF record field PRE.BATCH.ROUT. Such routine doesn’t have any parameters; it requires correspondent PGM.FILE record (type S).

If PGM.FILE record doesn’t exist, the job COB.EXECUTE.API fails, its status becomes 3 and the following error appears in its COMO:

COMO contents

```
_BNK/COB.INITIALISE_COB.EXECUTE.API_2_18 FEB 2015_16:16:27_Fatal error in
(COB.EXECUTE.API) error EB.RTN.SUBROUTINE.DOES.NOT.EXISTMY.COB.RTN contract:
```

Batch BNK/DATE.CHANGE (stage D000) changes the current work day in DATES so all of them are the same now.

As soon as all jobs finish, in TSA.STATUS record COB the field SERVICE.CONTROL is set to STOP again.

Start and end of COB are reflected in noinput fields association:

T24

```
 Model Bank R14          TSA.SERVICE, INPUT

     SERVICE........... COB
 ------------------------------------------------------------------------------
...
  15. 1 DATE........... 17 APR 2014
  16. 1 STARTED........ 16/05/2014 05:22:33
  17. 1 STOPPED........ 16/05/2014 06:02:09
  18. 1 ELAPSED........ 00:39:36
  19. 1 TRANSACTIONS... 121975
  15. 2 DATE........... 14 APR 2014
  16. 2 STARTED........ 16/05/2014 04:11:35
  17. 2 STOPPED........ 16/05/2014 04:49:32
  18. 2 ELAPSED........ 00:37:57
  19. 2 TRANSACTIONS... 119990
  15. 3 DATE........... 11 APR 2014
  16. 3 STARTED........ 16/05/2014 01:49:35
  17. 3 STOPPED........ 16/05/2014 02:26:12
  18. 3 ELAPSED........ 00:36:37
  19. 3 TRANSACTIONS... 119107
...
 ------------------------------------------------------------------------------
```

No special action (except backup and - most probably - setting of currency rates) is required to start the next work day.

## 1.69 More about locks

Returning to online work. As it was illustrated earlier, if one session holds a lock on some system-wide record, other users might experience problems.

Let’s lock F.LOCKING record FBNK.CUSTOMER using JED, like we did before, and click “new deal” in Browser, application CUSTOMER. Browser window with a rolling animated gif “hangs”... wait a little, then close it. Try again, close. Try again. See which locks were applied:

SHOW-ITEM-LOCKS output

```
    PORT        PID FILENAME          RECORDKEY         LOCK#         PORT/-PID

       1       1304 ..\bnk.data\eb\F_ FBNK.CUSTOMER     0x0425ba74,W        ---
                    LOCKING
    1006       3924 ..\bnk.data\eb\F_ FBNK.CUSTOMER       00000000,W     999999
                    LOCKING
    1007       2920 ..\bnk.data\eb\F_ FBNK.CUSTOMER       00000000,W     999999
                    LOCKING
```

Port 1 - JED. Port 1006 - one agent; port 1007 - another. 999999 - “waiting for a lock”. T24 actually doesn’t know anything about window being closed and if user clicks to the same place many times, many agents are started until a timeout is reached and user gets the error “Request has timed out”.

In some cases locks are not released and then the only solution is to log off problematic sessions, e.g.  for the example above:

jsh

```
LOGOFF 1006 1007 (V
```

Output

```
Port 1006 logged off
Port 1007 logged off
```

## 1.70 Enquiry - build routine

Another hook subroutine for ENQUIRY which is widely used is “build routine”.  It’s used to check and (if necessary) to amend user selection criteria.

Such tables as STMT.ENTRY are quite big and it’s a good idea to limit a selection that might probably take a lot of time (and possibly cause Browser timeout), as well as returning several millions of records that absolutely makes no sense to go through...

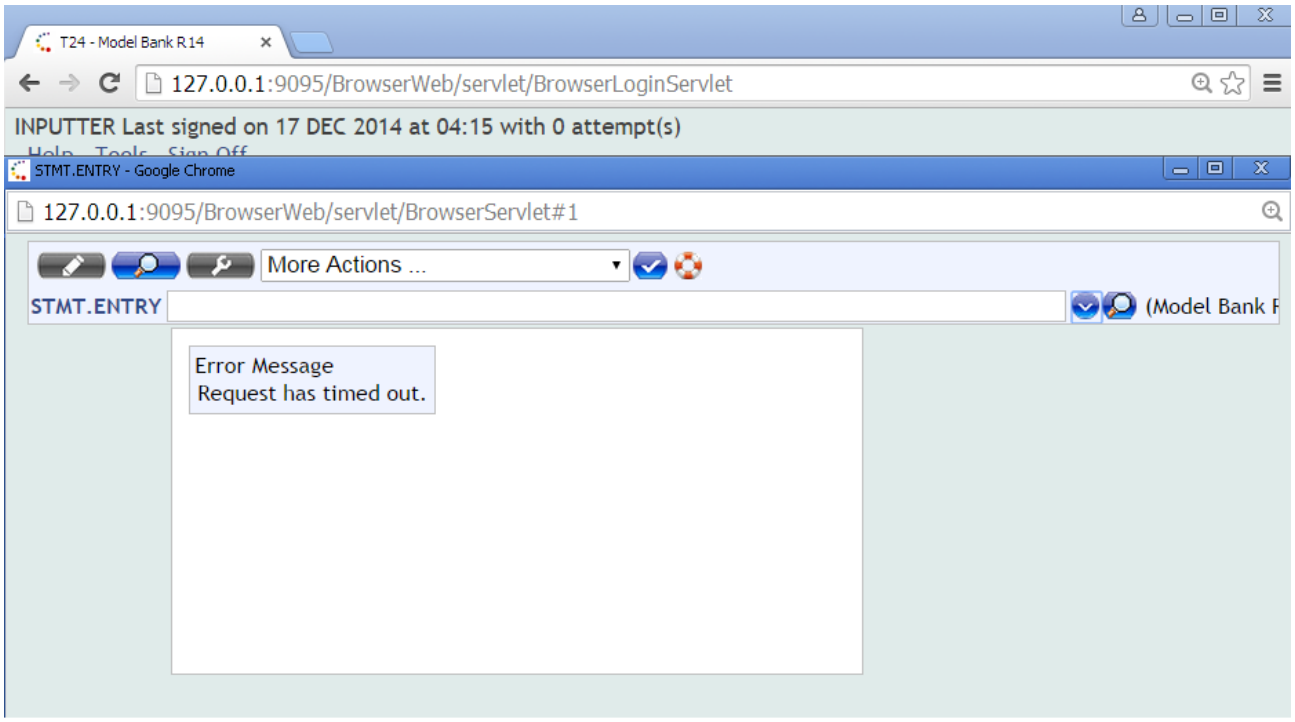


**Figure 42:  Timeout error after dropdown.**

Source code of “build routine”:

jBC

```
SUBROUTINE SUBR.ENQB(enq_data)
* T24 build routine for ENQUIRY
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_ENQUIRY.COMMON

   IF enq_data<2,1> EQ '' THEN   ;* no selections were input

*      ENQ.ERROR = 'PLEASE PROVIDE SELECTION'
* If we asked for any selection we could make it mandatory
* in ENQUIRY record...
* We'd rather retrieve 5 records using BASIC select and give them
* as the only output...

   fn_se = 'F.STMT.ENTRY'  ;  f_se = ''
   CALL OPF(fn_se, f_se)

   SELECT f_se TO id_list
   enq_data<2,1> = '@ID'
   enq_data<3,1> = 'EQ'

   FOR i = 1 TO 5
      READNEXT se_id FROM id_list THEN
         enq_data<4,1,-1> = se_id
      END ELSE ENQ.ERROR = 'ID retrieval error'  ;  RETURN
   NEXT i

   RETURN
END
```

Attach it to copied enquiry %STMT.ENTRY (it’s a good idea not to touch the initial one):

T24

```
 Model Bank R14          ENQUIRY, COPY

     ENQUIRY........... SE.TRAIN
 ------------------------------------------------------------------------------
   1 PAGE.SIZE ........ 4,19
   2 FILE.NAME......... STMT.ENTRY
   3. 1 FIXE
   4. 1 FIXED.SORT.....
   5. 1 OPEN.BRACKET...
   6. 1 SELECTION.FLDS.
   7. 1. 1 GB SEL.LABEL
   8. 1 SEL.FLD.OPER...
   9. 1 REQUIRED.SEL...
  10. 1 CLOSE.BRACKET..
  11. 1 REL.NEXT.FIELD.
  12. 1 BUILD.ROUTINE.. SUBR.ENQB                                    <=====
  13. 1. 1 HEADER......
  14. 1 FIELD.NAME..... @ID
...
```

In T24 supply command QUERY SE.TRAIN ...no, build routine wasn’t triggered.  Long SELECT can be however terminated by Ctrl-C:

T24

```
 ------------------------------------------------------------------------------
 17 DEC 2014 05:23:21  USER (22 APR) INPUTTER          [31929,ISELECTING
jQL Statement interrupted
Enter C to Continue, Q to Quit : _
```

After Q we see (spelling as of original):

T24

```
         No records were found that matched the selection criteria
         ERROR IN EXEXCUTION OF SELECT COMMAND
         SSELECT FBNK.STMT.ENTRY WITH   CO.CODE =
         "GB0010001"
```

At T24 command ENQUIRY SE.TRAIN all goes as expected and after pressing F5 without any selection almost instantly the following screen appears:

T24

```
 Model Bank R14


 ------------------------------------------------------------------------------
 1 169372905971841.020USDGBP140160001|       8USD    28 MAR 2999999-3.400301-000
 2 169374887972996.020GBPUSD140160001|   1.0|7GBP    31 MAR 2LD1408329142
 3 169376487080818.02020656                8USD    07 APR 2MM1408300020;5
 4 169378703262933.01069086                0USD    24 MAR 2FT14083BMFF7
 5 169389127000368.040GBP1280000040001   1.0|1GBP    08 APR 2101013-1
 ------------------------------------------------------------------------------
 17 DEC 2014 05:27:52  USER (22 APR) INPUTTER          [31929,IPAGE   1 >>>1>>>
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

Screen doesn’t look very nice (column positions not set properly) so go and see it in Browser...

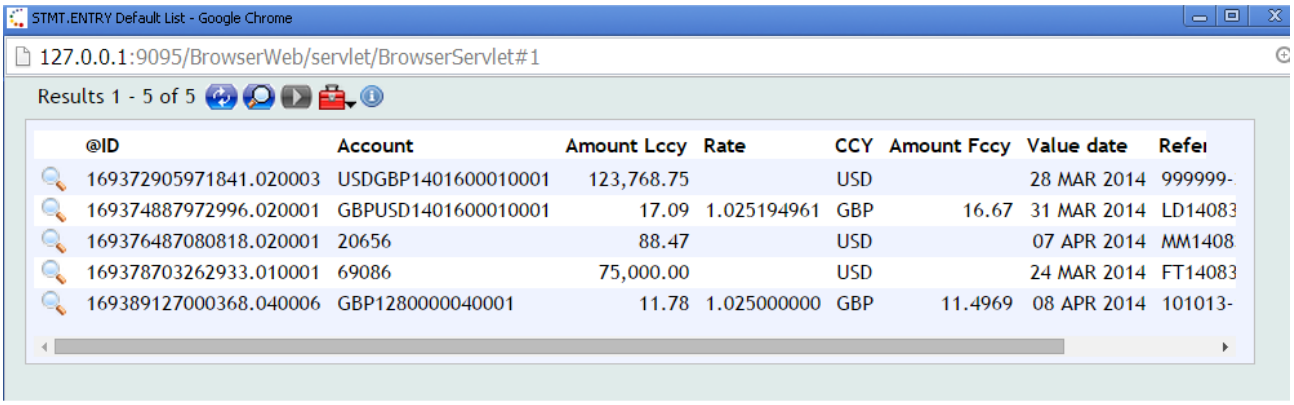


**Figure 43:  Limited enquiry output.**

Run this enquiry from OFS - again 5 records are returned:

tSS TAABS

```
ENQUIRY.SELECT,,INPUTT/123456,SE.TRAIN,
```

OFS output

```
,@ID::@ID/ACCOUNT.NUMBER::Account/AMOUNT.LCY::Amount Lccy/EXCHANGE.RATE::Rate/CURRENCY
::CCY/AMOUNT.FCY::Amount Fccy/VALUE.DATE::Value date/TRANS.REFERENCE::Reference,"16937
2905971841.020003                                                                 "
  "USDGBP1401600010"  "         123,768.75"   "       "       "USD"   "
    "   "28 MAR 2014"   "999999-3.400301-000      ","169374887972996.020001
                                                        "    "GBPUSD1401600010"
"              17.09"       "   1.02"       "GBP"   "             16.67"   "31 MAR 20
14"   "LD1408329142            ","169376487080818.020001
...
```

And supplying the selection criterion (the fastest way to show the result will be to select by @ID) cancels the retrieval of 5 records:

tSS TAABS

```
ENQUIRY.SELECT,,INPUTT/123456,SE.TRAIN,@ID:EQ:=169376487080818.020001
```

Of course we can fool the routine supplying something like...

OFS input

```
ENQUIRY.SELECT,,INPUTT/123456,SE.TRAIN,@ID:LK:=...
```

...but then the routine can be corrected to catch this.

## 1.71  Enquiry - start from a subroutine

Enquiry can also be started from a jBC routine.  In this case build routine isn't triggered as well as in QUERY command though all necessary logic can be done inside the launching routine.

Example:

jBC

```
SUBROUTINE SUBR.0
* T24 mainline routine
$INSERT I_COMMON
$INSERT I_EQUATE

   enq_data<1> = '%ACCOUNT'
   enq_data<2,1> = '@ID'
   enq_data<3,1> = 'LK'
   enq_data<4,1,1> = 'USD...'

   CALL ENQUIRY.DISPLAY(enq_data)

   RETURN
END
```

Run it and see the default list of internal accounts in USD currency:

T24

```
 Model Bank R14


 ------------------------------------------------------------------------------
 1             U|             Implement| Cash            USD       -45,00      10001
 2             U|             Implement| Cash            USD        -2,00      10001
 3             U|             Implement| Cash            USD        -2,00      10001
 4             U|             Implement| Cash            USD                   10001
 5             U|  USDTILL1   Treasury   Cash            USD                   10001
 6             U|  USDTILL2   Treasury   Cash            USD                   10001
 7             U|             Implement| Cash            USD   -63,684,47      10001
 8             U|             Customer | Cash            USD      -201,00      10001
 9             U|  USDTILLVA  Treasury   Cash            USD                   10001
10   USD100110001  VAULT10011 Implement| Vault           USD                   10011
11             U|  USD1       Treasury   Vault           USD                   10011
12             U|  USD2       Treasury   Vault           USD                   10011
13             U|  USDVA      Treasury   Vault           USD                   10011
14   USD100900001  TRAVE10090 Implement| TC Purchased    USD                   10090
15             U|  USDTCPUR1  Treasury   CashCheqClrg    USD                   10099
16             U|  USDTCPUR2  Treasury   CashCheqClrg    USD                   10099
17-            U|--USDTCPURVA-Treasury  -CashCheqClrg   -USD--                10099
187  USD1020100012:CHEQU102012Implement|UCheqPaid - Coll7USD,I PAGE   1 >>>3>>>
19C  USD109050001  TCSTO10905 Imple
9AWAITING PAGE INSTRUCTIONS
```

## 1.72 JBoss and BrowserWeb.war settings of interest

JAVA_HOME environment variable has to be set up for JBoss to work:

jsh or cmd.exe

```
echo %JAVA_HOME%
```

Output

```
D:\Temenos\ModelBank-R14\T24\Env\..\..\Infra\Java
```

Memory for JVM - see (JBoss)\bin\run.conf.bat:

run.conf.bat

```
set "JAVA_OPTS=-Xms1024M -Xmx2048M -XX:MaxPermSize=512M"
```

Too much memory given to JVM slows down things, too little may cause the error like:

JBoss log

```
2014-08-20 17:50:11,983 ERROR [org.jboss.ejb.plugins.LogInterceptor](WorkManager(2)-5)
Unexpected Error in method: public abstract void javax.jms.Mess$
java.lang.OutOfMemoryError: Java heap space
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    at com.jbase.jremote.io.PrefixByteBuffer.resize(PrefixByteBuffer.java:106)
    at com.jbase.jremote.io.PrefixByteBuffer.readFrom(PrefixByteBuffer.java:175)
```

Host and port to connect to - see (JBoss)\server\default\deploy\t24-ds.xml:

t24-ds.xml

```
<config-property name="hosts" type="java.lang.String">127.0.0.1</config-property>
<config-property name="ports" type="java.lang.String">20014</config-property>
```

OFS.SOURCE to use - same file:

t24-ds.xml

```
<config-property name="envVariables" type="java.lang.String">OFS_SOURCE=BROWSERTC
   </config-property>
```

Character set - also there:

t24-ds.xml

```
<config-property name="charset" type="java.lang.String">UTF-8</config-property>
```

Timeouts, again in the same file:

t24-ds.xml

```
<config-property name="actionTimeout" type="java.lang.Integer">120</config-property>
...
<idle-timeout-minutes>15</idle-timeout-minutes>
```

Last thing to look at in this file - minimum and maximum size of connections pool:

t24-ds.xml

```
<min-pool-size>6</min-pool-size>
<max-pool-size>10</max-pool-size>
```

*(Corresponding numbers of agents will be launched at application server, for the example above 6 agents will always be active - dropping to 2-3 sometimes - and should all 10 agents be busy (or in an infinite loop or debugger), no new requests would pass through.)*

Connection method - see file BrowserParameters.xml in (JBoss)\server\default\ deploy\BrowserWeb.war (might be located in other directory though, depends on BrowserWeb.war deployment method:

BrowserParameters.xml

```
<parameterName>Server Connection Method</parameterName>
<parameterValue>AGENT</parameterValue>
<!-- Options: INSTANCE / AGENT / JMS / AGENT_JREMOTE  -->
```

Never use JMS option for T24 - several attempts at different projects couldn’t make it work properly.  AGENT is OK.

## 1.73 jBC functions

Besides subroutines, we can write and use functions in jBC. Let’s make a function from the expression we used earlier in EVAL (see CATEGORY chapter - getting category name).

Firstly, write the function code, use CATG.NAME file in ETC.BP:

jBC

```
FUNCTION CATG.NAME(catg_code)

RETURN (OCONV(catg_code, 'TF.CATEGORY;V1;1;1'))
```

Then the program that uses this function.  Function has to be defined in its code with the DEFFUN statement.

The program source code:

jBC

```
DEFFUN CATG.NAME()

   EXECUTE 'SELECT F.CATEGORY SAMPLE 5' :@FM: 'SAVE-LIST CATG-LIST'
   GETLIST 'CATG-LIST' TO catg_list ELSE
      CRT 'GETLIST error'
      STOP
   END

   LOOP
      REMOVE catg FROM catg_list SETTING the_stat
      GOSUB GET.NAME
      IF the_stat EQ 0 THEN BREAK   ;* no more entries
   REPEAT

   catg = 100000    ;* test non-existent one
   GOSUB GET.NAME

   STOP

GET.NAME:
   CRT SQUOTE(CATG.NAME(catg))
   RETURN
END
```

Compile both source files and run the program:

Program output

```
 5 Records selected

5 record(s) saved to list 'CATG-LIST'
'*** Other Client Accounts'
'IAS FX Unreal Profit Reserve'
'***Not Used***'
'*** Margin Accounts'
'Savings Account'
''
```

If we omit DEFFUN statement, compilation error comes up:

Compiler output

```
jsh r14  -->BASIC ETC.BP PROG.1
PROG.1
[error 1        (68)] "PROG.1", 21 (offset 1) :
Variable CATG.NAME was assumed to be a DIMensioned array but no DIM statement was found
jbccom -f -d -aETC.BP -IETC.BP BASIC_3.b failed , command returned a code of 1
jcompile: BASIC_3.j deleted
jcompile: BASIC_3.c deleted
jcompile: Returned an error code of 8
 ** Unable to compile source PROG.1 **
```

Compiler, seeing parentheses notation, considers CATG.NAME to be a dimensioned array.

## 1.74 First local application

Let it be the simplest one - L-type. We'll try to store there the VERSION information whenever user edits an application record, so far for ACCOUNT (but keeping in mind other applications as well). This information might be useful later for analysis or audit purposes.

Step 1. Copy the minimum set of templates:

jsh

```
COPY FROM T24.BP TO ETC.BP TEMPLATE
COPY FROM T24.BP TO ETC.BP TEMPLATE.FIELDS
```

Step 2. Rename them to chosen application name, e.g. VERS.INFO:

jsh

```
COPY FROM ETC.BP TEMPLATE,VERS.INFO DELETING
COPY FROM ETC.BP TEMPLATE.FIELDS,VERS.INFO.FIELDS DELETING
```

*(DELETING keyword deletes the initial files - TEMPLATE and TEMPLATE.FIELDS - after copying.)*

Step 3. Edit templates:

VERS.INFO

```
SUBROUTINE VERS.INFO
*-----------------------------------------------------------------------
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_Table
*-----------------------------------------------------------------------
   Table.name = 'VERS.INFO'
   Table.title = 'VERSION information'       ;* Screen title
   Table.stereotype = 'L'                    ;* H, U, L, W or T
   Table.product = 'EB'
   Table.subProduct = ''
   Table.classification = 'INT'        ;* As per FILE.CONTROL
   Table.systemClearFile = 'Y'         ;* As per FILE.CONTROL
   Table.relatedFiles = ''             ;* As per FILE.CONTROL
   Table.isPostClosingFile = ''        ;* As per FILE.CONTROL
   Table.equatePrefix = 'V.INF'        ;* Use to create I_F...
   Table.idPrefix = ''       ;* Used by EB.FORMAT.ID if set
   Table.blockedFunctions = ''   ;* Space delimited list of blocked funcs
   Table.trigger = ''
*-----------------------------------------------------------------------
   RETURN
END
```

The syntax above looks like OOP but it’s not it.  Table.name, Table.title etc are just variables defined in insert file I_Table located in T24.BP:

I_Table contents

```
COMMON/OBJECT.TEMPLATE/Table.fieldNeighbourArray(C$SYSDIM),Table.name,
   Table.title,
   Table.stereotype,
   Table.product,
...
```

Fields definition:

VERS.INFO.FIELDS

```
   SUBROUTINE VERS.INFO.FIELDS
*-----------------------------------------------------------------------------
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_DataTypes
*-----------------------------------------------------------------------------
* ID structure: APPLICATION*RECORD_ID*TIMESTAMP
   CALL Table.defineId('VERS.INFO.ID', T24_String)
*-----------------------------------------------------------------------------
   CALL Table.addFieldDefinition('OPERATOR', '64', 'A', '')
   CALL Field.setCheckFile('USER')
   CALL Table.addFieldDefinition('VERSION', '64', 'A', '')
   CALL Field.setCheckFile('VERSION')
*-----------------------------------------------------------------------------
   RETURN
*-----------------------------------------------------------------------------
END
```

Compile both subroutines.

Step 3.  In T24 launch the application EB.DEV.HELPER with record @ID = our application name.  Populate the following fields and commit:

T24

```
 Model Bank R14          Development Helper INPUT

     APPLICATION.ID.... VERS.INFO
 ------------------------------------------------------------------------------
   1 PGM.FILE.......... YES
   2 FILE.CONTROL...... YES
   3 INSERT............ YES
   4 CREATE.FILE....... YES
   5 UPDATE.SS......... YES
   6 CREATE.DAS.PGMS...
```

Then open this record with V function (in some cases V function alone can be used right from the start), commit:

T24

```
--------------------------------------------------------------------------------
06 FEB 2015 11:51:54  USER (22 APR) VLADIMIR.K        [25084,ITXN COMPLETE
ACTION _
CONTINUE (Y)                                          Created PGM.FILE record
```

...

T24

```
--------------------------------------------------------------------------------
06 FEB 2015 11:51:54  USER (22 APR) VLADIMIR.K        [25084,ITXN COMPLETE
ACTION _
CONTINUE (Y)                                    Created FILE.CONTROL record
```

...

T24

```
--------------------------------------------------------------------------------
06 FEB 2015 11:51:54  USER (22 APR) VLADIMIR.K        [25084,ITXN COMPLETE
ACTION _
CONTINUE (Y)                                                  Created insert
```

...

T24

```
--------------------------------------------------------------------------------
06 FEB 2015 11:51:54  USER (22 APR) VLADIMIR.K        [25084,ITXN COMPLETE
ACTION _
CONTINUE (Y)                                                   Created file.
[ 417 ] File ..\bnk.data\eb\F_VERS_INFO created , type = JR
```

*(“File already exists.  File not created” errors can be ignored.)*

T24

```
--------------------------------------------------------------------------------
06 FEB 2015 11:56:09  USER (22 APR) VLADIMIR.K        [25084,IN]
ACTION _
CONTINUE (Y)                                 Update STANDARD.SELECTION record
```

That’s it.

We can now input “VERS.INFO L” command at “AWAITING APPLICATION” prompt:

T24

```
Model Bank R14           VERS.INFO - Default List

   ID                                                                    FUNCT.
------------------------------------------------------------------------------

         No records were found that matched the selection criteria

         SSELECT F.VERS.INFO









------------------------------------------------------------------------------
06 FEB 2015 11:56:51  USER (22 APR) VLADIMIR.K        [25084,IPAGE   1 >>>1>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

See that SS record has been created...

T24

```
Model Bank R14          STANDARD SELECTION FIELDS SEE

    FILE.NAME......... VERS.INFO
------------------------------------------------------------------------------
  1. 1 SYS.FIELD.NAME. @ID
  2. 1 SYS.TYPE....... D
  3. 1. 1 SYS.FIELD.NO 0
  4. 1. 1 SYS.VAL.PROG IN2A
  6. 1 SYS.DISPLAY.FMT 35L
  7. 1 SYS.ALT.INDEX.. N
 10. 1 SYS.SINGLE.MULT S
 11. 1 SYS.LANG.FIELD. N
 12. 1 SYS.GENERATED.. Y
  1. 2 SYS.FIELD.NAME. VERS.INFO.ID
  2. 2 SYS.TYPE....... D
  3. 2. 1 SYS.FIELD.NO 0
  4. 2. 1 SYS.VAL.PROG IN2A
  6. 2 SYS.DISPLAY.FMT 35L
  7. 2 SYS.ALT.INDEX.. N
 10. 2 SYS.SINGLE.MULT S
 11. 2 SYS.LANG.FIELD. N
 12. 2 SYS.GENERATED.. Y
  1. 3 SYS.FIELD.NAME. OPERATOR
  2. 3 SYS.TYPE....... D
  3. 3. 1 SYS.FIELD.NO 1
  4. 3. 1 SYS.VAL.PROG IN2A
```

```
   6. 3 SYS.DISPLAY.FMT 64L
   7. 3 SYS.ALT.INDEX.. N
  10. 3 SYS.SINGLE.MULT S
  11. 3 SYS.LANG.FIELD. N
  12. 3 SYS.GENERATED.. Y
  14. 3. 1 SYS.REL.FILE USER               Contains the SMS details of each u|
   1. 4 SYS.FIELD.NAME. VERSION
   2. 4 SYS.TYPE....... D
   3. 4. 1 SYS.FIELD.NO 2
   4. 4. 1 SYS.VAL.PROG IN2A
   6. 4 SYS.DISPLAY.FMT 64L
   7. 4 SYS.ALT.INDEX.. N
  10. 4 SYS.SINGLE.MULT S
  11. 4 SYS.LANG.FIELD. N
  12. 4 SYS.GENERATED.. Y
  14. 4. 1 SYS.REL.FILE VERSION            Enable Users to redesign the Data |
  35 CURR.NO........... 3
  36. 1 INPUTTER....... 54_VLADIMIR.K__OFS_EB.DEV.HELPER
  37. 1 DATE.TIME...... 06 FEB 15 12:13
  38 AUTHORISER........ 54_VLADIMIR.K_OFS_EB.DEV.HELPER
  39 CO.CODE........... GB-001-0001        Model Bank R14
  40 DEPT.CODE......... 1                  Implementation
 ------------------------------------------------------------------------------
```

...as well as file dictionary:

jsh

```
LIST DICT F.VERS.INFO
```

Output

```
DICT F.VERS.INFO

@ID
VERS.INFO.ID
OPERATOR
VERSION
GROUP.MAP
@
XBUILD.DATEX
VERS_INFO@XMLMAP

 8 Records Listed
```

jsh

```
CT DICT F.VERS.INFO OPERATOR
```

Output

```
   OPERATOR
001 D
002 1
003
004 OPERATOR
005 32L
006 S
007
...
```

PGM.FILE record:

T24

```
PROGRAM            VERS.INFO
-------------------------------------------------------------------------------
  1 TYPE.............. L
  2. 1 GB SCREEN.TITLE VERSION information
  5 PRODUCT........... EB
 26 CURR.NO........... 1
 27. 1 INPUTTER....... 4122_VLADIMIR.K
 28. 1 DATE.TIME...... 06 FEB 15 12:13
 29 AUTHORISER........ 4122_VLADIMIR.K
 30 CO.CODE........... GB-001-0001         Model Bank R14
 31 DEPT.CODE......... 1                   Implementation
```

FILE.CONTROL record:

T24

```
    FILE.............. VERS.INFO
-------------------------------------------------------------------------------
  1. 1 DESCRIPTION.... VERSION information
  2 PRODUCT........... EB                  System Core
  4 FILE TYPE......... 3
  5 FILE MODULO....... 11
  6 CLASSIFICATION.... INT
  8 SYS.CLEAR.FILES... Y
  9 CUS.CLEAR.FILES... N
 20 CURR.NO........... 1
 21. 1 INPUTTER....... 4122_VLADIMIR.K
 22. 1 DATE.TIME...... 06 FEB 15 12:13
 23 AUTHORISER........ 4122_VLADIMIR.K
 24 CO.CODE........... GB-001-0001         Model Bank R14
 25 DEPT.CODE......... 1                   Implementation
```

Finally, insert file (located in BP):

jsh

```
CT BP I_F.VERS.INFO
```

Output

```
001 * I_F.VERS.INFO Created 06 FEB 15 at 12:30PM by r14/VLADIMIR.K
002 *      PREFIX[V.INF]
003                       EQU V.INF.OPERATOR TO 1,   VersInfo_Operator TO 1,
004                           V.INF.VERSION TO 2,   VersInfo_Version TO 2
```

To be able to use this insert file in local develoments we need to move it from BP to ETC.BP:

jsh

```
COPY FROM BP TO ETC.BP I_F.VERS.INFO DELETING
```

## 1.75 Population of L-type application

The application developed in the previous chapter can’t be populated manually in T24. Firstly let’s try to populate it in JED and see how the resulting record looks in T24:

JED F.VERS.INFO ACCOUNT*14378*123456

```
0001 INPUTTER
0002 ACCOUNT,TRAIN
```

*(Timestamp for @ID - 123456 - is just for test.)*

Save the record, see it in T24:

T24

```
    VERS.INFO.ID..... ACCOUNT*14378*123456
 ------------------------------------------------------------------------------
   1 OPERATO INPUTTER            INPUTTER
   2 VERSION ACCOUNT,TRAIN                                  EB-MISSING.RECORD
```

2 things we see right now: both fields length (64) is too big to display full field name in terminal mode; also VERSION record that we input manually doesn’t exist.

Let’s try to correct the former issue at least for the field OPERATOR. In SS record for application USER we can see that the length of user @ID is 16. We can be on the safe side (assuming possible core changes in the future) to set it to 32 in our application:

Change in the source code of VERS.INFO.FIELDS

```
CALL Table.addFieldDefinition('OPERATOR', '32', 'A', '')
```

After recompilation all we need is to rebuild SS record in EB.DEV.HELPER.

The record is seen now as:

T24

```
    VERS.INFO.ID...... ACCOUNT*14378*123456
 ------------------------------------------------------------------------------
   1 OPERATOR.......... INPUTTER            INPUTTER
   2 VERSION ACCOUNT,TRAIN                                    EB-MISSING.RECORD
```

This record can be deleted from file to keep it tidy:

jsh

```
DELETE F.VERS.INFO ACCOUNT*14378*123456
```

...or - without typing the @ID:

jsh

```
SELECT F.VERS.INFO

 1 Records selected

>DELETE F.VERS.INFO
```

To populate the application the way it was intended we create the “input routine” and attach it to application VERSION.CONTROL, record ACCOUNT.

Subroutine source:

jBC - ADD.VERS.INFO

```
SUBROUTINE ADD.VERS.INFO
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_F.VERS.INFO

   fn_info = 'F.VERS.INFO'  ;  f_info = ''
   CALL OPF(fn_info, f_info)

   time_stamp = ''
   CALL ALLOCATE.UNIQUE.TIME(time_stamp)   ;* core T24 subroutine

   the_id = APPLICATION : '*' : ID.NEW : '*' : time_stamp
   the_rec = ''
   the_rec<VersInfo_Operator> = OPERATOR
   the_rec<VersInfo_Version> = APPLICATION : PGM.VERSION

   CALL F.WRITE(fn_info, the_id, the_rec)

   RETURN
END
```

While constructing the record for our local application we always address fields via their names in insert file rather than their numbers.  If in the future the

order of fields changes, all we need is to recompile the source code using the new insert file.

VERSION.CONTROL:

T24

```
    PGM.NAME.VER.TYPE. ACCOUNT
------------------------------------------------------------------------------
  1. 1 AUTO.FIELD.NAME
  2. 1 AUTO.OLD.CONT..
  3. 1 AUTO.FIELD.RTN.
  4. 1 FIELD.NAME.....
  5. 1 VALIDATION.RTN.
  6. 1 INPUT.RTN...... ADD.VERS.INFO
```

Edit the ACCOUNT record; VERSION has to have the field EXC.INC.RTN set to YES to comply to VERSION.CONTROL settings:

T24

```
Model Bank R14          ACCOUNT,AF.INPUT INPUT
                                  CAD  Current Account
    ACCOUNT.NUMBER.... 66575              THOMRE
------------------------------------------------------------------------------
Customer........ 188919     Thomson Reuters
Category........ 1-001  Current Account
Currency........ CAD Canadian Dollar
Mnemonic........ THOMRECAD
Short Title..... THOMRE
Account Title 1. THOMRE
Account Title 2. TEST                                          <===== here
Account Officer. 62   Corporate Loan|
```

Commit, see VERS.INFO L:

T24

```
   ID                                  OPERATOR                        FUNCT.
------------------------------------------------------------------------------
1  ACCOUNT*66575*1056448428.00         VLADIMIR.K
```

And the record contents:

T24

```
    VERS.INFO.ID...... ACCOUNT*66575*1056448428.00
------------------------------------------------------------------------------
  1 OPERATOR.......... VLADIMIR.K          VLADIMIR K
  2 VERSION ACCOUNT,AF.INPUT
```

More changes, see results:

jsh

```
LIST F.VERS.INFO
```

Output

```
@ID.......... ACCOUNT*21067*1650248908.00
@ID.......... ACCOUNT*21067*1650248908.00
VERS.INFO.ID. ACCOUNT*21067*1650248908.00
OPERATOR..... VLADIMIR.K
VERSION...... ACCOUNT,

@ID.......... ACCOUNT*66575*1056448428.00
@ID.......... ACCOUNT*66575*1056448428.00
VERS.INFO.ID. ACCOUNT*66575*1056448428.00
OPERATOR..... VLADIMIR.K
VERSION...... ACCOUNT,AF.INPUT

@ID.......... ACCOUNT*66575*1650248972.00
@ID.......... ACCOUNT*66575*1650248972.00
VERS.INFO.ID. ACCOUNT*66575*1650248972.00
OPERATOR..... VLADIMIR.K
VERSION...... ACCOUNT,

 3 Records Listed
```

We can add the information about what exactly was changed; thus we would be able to track changes even for unauthorised records.

## 1.76 Delivery overview

Delivery is the huge theme and just an overview can only fit here. It’s one of the oldest T24 (read: Globus) modules; a lot of changes in the processing hence too many obsolete things that are still there. Hope this example of setup for SWIFT channel is correct.

**Outward delivery**.

DE.CARRIER:

T24

```
Model Bank R14          DELIVERY CARRIER PARAMETERS INPUT

    CARRIER.......... SWIFT
--------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. Standard SWIFT carrier definition
  2 ADDRESS........... SWIFT
  3 FORMAT.MODULE..... SWIFT
  4 CARRIER.MODULE.... GENERIC
  5 INTERFACE......... MODEL               MODEL INTERFACE
  6 IN.FORMAT.MODULE.. SWIFT
  7 IN.BYTE.LEN.......
  8 RESERVED.8........
  9 RESERVED.7........
 10 RESERVED.6........
 11 RESERVED.5........
 12 RESERVED.4........
 13 RESERVED.3........
 14 RESERVED.2........
 15 RESERVED.1........
 16. 1 LOCAL.REF......
--------------------------------------------------------------------------------
16 FEB 2015 06:11:35  USER (22 APR) VLADIMIR.K        [4178,INPAGE 1   >>>2>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Field 5 refers to DE.INTERFACE:

T24

```
Model Bank R14          DELIVERY INTERFACE PARAMETERS SEE

    INTERFACE......... MODEL
--------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. MODEL INTERFACE
  2 HARDWARE..........
  3 PROTOCOL..........
  4 NETWORK...........
  5 LINE.COMM.........
  6 TIME.OUT.SECS.....
  7. 1 FILE.PATHNAME..
  8 OUT.IF.ROUTINE....
  9 IN.IF.ROUTINE.....
 10 ACK.REQUIRED...... Y
 11 AUDIT.SEQ.CHECKING
 12 STARTUP.ROUTINE...
 13 SHUTDOWN.ROUTINE..
 14 ACK.ROUTINE.......
 15 RESERVED.7........
 16 RESERVED.6........
--------------------------------------------------------------------------------
18 FEB 2015 05:29:32  USER (22 APR) VLADIMIR.K        [1410,INPAGE 1   >>>2>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Local routine can be attached to fields OUT.IF.ROUTINE and IN.IF.ROUTINE to provide customized processing of outgoing and incoming messages respectively.

Setup of message fields - DE.MESSAGE; example for MT103, page 1 of 23:

T24

```
Model Bank R14          DE.MESSAGE SEE

    MESSAGE.TYPE...... 103
------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. CUSTOMER CREDIT TRANSFER
  2 COPIES............ Y
  3 TRANSLATION....... Y
  4 DELETE............ Y
  5 TEST.KEY.REQ...... NO
  7. 1 FIELD.NAME..... SENDER REF
  8. 1 LENGTH......... 16
  9. 1 PRINT.TYPE..... A
 10. 1 SINGLE.MULTI... S
 11. 1 MANDATORY...... Y
  7. 2 FIELD.NAME..... TIME IND
  8. 2 LENGTH......... 8
  9. 2 PRINT.TYPE..... A
 10. 2 SINGLE.MULTI... S
 11. 2 MANDATORY...... NO
  7. 3 FIELD.NAME..... OPERATION CODE
------------------------------------------------------------------------------
16 FEB 2015 06:18:11  USER (22 APR) VLADIMIR.K        [4178,INPAGE 1   >>23>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Sequence and format of tags - DE.FORMAT.SWIFT, page 1 of 27:

T24

```
Model Bank R14          DE.FORMAT.SWIFT INPUT

    ID.............. 103.1.1             CUSTOMER CREDIT TRANSFER
------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. CUSTOMER CREDIT TRANSFER
  2 MESSAGE TYPE...... 103
  3. 1 FIELD TAG...... 20
  4. 1 FIELD NAME..... SENDER REF
  5. 1. 1 CONVERSION..
  6. 1 CALCULATION....
  7. 1 MULTIPLES...... NO
  8. 1 INWARD ROUTINE. NO
  9. 1 OUTWARD ROUTINE NO
 10. 1 HEADER NAME.... TRANS.REF
  3. 2 FIELD TAG...... 13C
  4. 2 FIELD NAME..... TIME IND
  5. 2. 1 CONVERSION.. TIME.CET
  6. 2 CALCULATION....
  7. 2 MULTIPLES...... Y
  8. 2 INWARD ROUTINE. NO
------------------------------------------------------------------------------
```

Correspondent DE.MAPPING record (FT-related one), page 1 of 24:

T24

```
Model Bank R14          DE.MAPPING INPUT

    ID.............. 103.FT.1            CUSTOMER CREDIT TRANSFER
------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. CUSTOMER CREDIT TRANSFER
  2. 1 INPUT.REC.NO... 1
  3. 1 INPUT.REC.DESC. FUNDS TRANSFER
  4. 1 INPUT.FILE..... FUNDS.TRANSFER
  2. 2 INPUT.REC.NO... 2
  3. 2 INPUT.REC.DESC. MISCELLANEOUS DETAILS
  4. 2 INPUT.FILE.....
  5. 1 INPUT.POSITION. 2.2
  6. 1 INPUT.NAME.....
  7. 1. 1 FIELD.DESCR.
  8. 1 FIELD.NAME..... SENDER REF
  9. 1 HEADER.NAME.... TRANS.REF
  5. 2 INPUT.POSITION. 2.39
  6. 2 INPUT.NAME.....
  7. 2. 1 FIELD.DESCR.
  8. 2 FIELD.NAME..... OPERATION CODE
------------------------------------------------------------------------------
16 FEB 2015 06:24:07  USER (22 APR) VLADIMIR.K        [4178,INPAGE 1   >>24>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

The field of interest is:

T24

```
 16 ROUTINE...........
```

Here a hook routine can be attached to amend - if necessary - message mapping.  Mapping is the first stage of outward message processing, it now happens at the time of authorising the deal (e.g.  funds transfer).  In older times there was an additional “phantom” process to do that.

Mapping takes several (up to 9) records, starting from the deal itself, and creates F.DE.O.HANDOFF record, e.g.:

jsh

```
SELECT F.DE.O.HANDOFF WITH LINE1 LIKE 103... SAMPLE 1

 1 Records selected

>CT F.DE.O.HANDOFF
```

Output

```
   D20140516375402420604
001 103.FTOT.1*20140422.OT03
002 36293
003
004 1
005 USD
006
007 20140422
008
009
010
011 20656
...
```

There is no application DE.O.HANDOFF so this file can be seen only from jsh.  Too illustrate the mapping we can use the core enquiry DE.HANDOFF.DETS.

Selection stage:

T24

```
Model Bank R14          ENQUIRY INPUT

    SELECT NAME....... DE.HANDOFF.DETS
--------------------------------------------------------------------------------
  1. 1 SELECTION.TYPE. <k>
  2. 1 SELECTION.FIELD DELIVERY.REF
  3. 1 OPERAND........ EQ
  4. 1. 1 LIST........ D20140516375402420604           <===== F.DE.O.HANDOFF @ID
  1. 2 SELECTION.TYPE. <d>
  2. 2 SELECTION.FIELD MAPPING.KEY
  3. 2 OPERAND........
  4. 2. 1 LIST........
  1. 3 SELECTION.TYPE. <d>
  2. 3 SELECTION.FIELD BANK.DATE
  3. 3 OPERAND........
  4. 3. 1 LIST........
  1. 4 SELECTION.TYPE. <d>
  2. 4 SELECTION.FIELD MACHINE.DATE
  3. 4 OPERAND........
  4. 4. 1 LIST........
--------------------------------------------------------------------------------
16 FEB 2015 06:43:23  USER (22 APR) VLADIMIR.K        [19934, PAGE 1   >>>3>>>
ACTION
```

Result:

T24

```
Model Bank R14          Delivery Handoff Details

  Position Field name               Field Content
--------------------------------------------------------------------------------
1 0.0      DELIVERY KEY             D20140516375402420604
2 0.1      MAPPING.KEY              103.FTOT.1
3 0.2      BANK.DATE                20140422
4 1.1       TRANS TYPE              OT03
5 1.2                               36293
6 1.3
7 1.4                               1
8 1.5                               USD
9 1.6
10 1.7                              20140422
11 1.8
12 1.9
13 1.10
14 1.11     HDR - ACCOUNT           20656
15 1.12                             1
16 1.13     CURRENCY                USD
--------------------------------------------------------------------------------
16 FEB 2015 06:43:39  USER (22 APR) VLADIMIR.K        [19934,IPAGE   1 >>>3>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

From the DE.FORMAT.SWIFT and DE.MAPPING screenshots above we can see that in MT103 message for FT tag 20 (SENDER REF) is taken from the field 2 of record 2 of hand-off record.  See it at the page 14:

T24

```
Model Bank R14          Delivery Handoff Details

  Position Field name               Field Content
--------------------------------------------------------------------------------
1 1.204
2 1.205
3 2.1      HDR - CUSTOMER           100472
4 2.2      SENDER REF               FT141124H6HC                          <=====
5 2.3      HDR - LANGUAGE           1
6 2.4                               USD
7 2.5                               1
8 2.6                               1
9 2.7                               ::OUTREMITCHG
10 2.8                              USD
11 2.9                              20.00
12 2.10
13 2.11
14 2.12
15 2.13                             9000.00
16 2.14                             1
--------------------------------------------------------------------------------
```

*(Big records are read not fully so page 16 isn’t the last one.  If we move one page down, there will be “PAGE 15 >>17>>>” at the bottom.  To go to the last page enter command P9999.)*

The mapping can be customized by mentioned above DE.MAPPING routine.  For example, to add something to SENDER REF, put the following code there:

jsh

```
SUBROUTINE CORR.FT.REF(MAT HANDOFF.REC, ERR.MSG)
   HANDOFF.REC(2)<2> = 'BANK' : HANDOFF.REC(2)<2>[12]   ;* allowed length is 16
   RETURN                                               ;*  (see DE.MESSAGE)
END
```

*(INSERT of I_COMMON and I_EQUATE isn’t necessary - we don’t use any of the global variables declared there.)*

After attaching this routine to DE.MAPPING take the FT that we were looking into and copy it; commit the deal:

T24

```
Model Bank R14          FUNDS.TRANSFER, COPY                 REF FT141121Q8SD


--------------------------------------------------------------------------------


                                          TRANS TYPE = OT


                                          900.1.1      DEBIT ADVICE
                                          103.1.1      CABLE TO CUST









--------------------------------------------------------------------------------
17 FEB 2015 06:07:40  USER (22 APR) VLADIMIR.K        [25984, TXN COMPLETE
ACTION _
AWAITING ID
```

Handoff details, page 14:

T24

```
Model Bank R14          Delivery Handoff Details

  Position Field name               Field Content
-------------------------------------------------------------------------------
1 1.202                             GB0010001
2 1.203     HDR - DEPARTMENT        1
3 1.204
4 1.205
5 2.1       HDR - CUSTOMER          100472
6 2.2       SENDER REF              BANKFT141121Q8SD                      <=====
7 2.3       HDR - LANGUAGE          1
8 2.4                               USD
9 2.5                               1
10 2.6                              1
11 2.7                              ::OUTREMITCHG
12 2.8                              USD
13 2.9                              5.00
14 2.10
15 2.11
16 2.12
-------------------------------------------------------------------------------
17 FEB 2015 06:12:17  USER (22 APR) VLADIMIR.K        [25984,IPAGE  14 >>16>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Back to initial deal.  After mapping the message is formatted and delivered (if all is correct) by online service BNK/SWIFT.OUT (should have AUTO in the field 6 during regular work day):

T24

```
Model Bank R14          TSA.SERVICE SEE

    SERVICE........... BNK/SWIFT.OUT
-------------------------------------------------------------------------------
  1. 1 DESCRIPTION.... DE FORMATTING SERVICE
  3. 1 WORK.PROFILE... DE.FORMATTING.SERVICE USED FOR DE FORMATTING SERVICE
  5 USER.............. INPUTTER            INPUTTER
  6 SERVICE.CONTROL... STOP
  7 REVIEW.TIME....... 10
 15. 1 DATE........... 22 APR 2014
 16. 1 STARTED........ 06/06/2014 08:25:08
 17. 1 STOPPED........ 06/06/2014 08:34:51
 18. 1 ELAPSED........ 00:09:43
 15. 2 DATE........... 22 APR 2014
 16. 2 STARTED........ 23/05/2014 15:30:36
 17. 2 STOPPED........ 29/05/2014 12:02:34
 18. 2 ELAPSED........ 140:31:58
 15. 3 DATE........... 19 MAR 2014
 16. 3 STARTED........ 08/04/2014 16:49:31
 17. 3 STOPPED........ 21/05/2014 13:19:13
-------------------------------------------------------------------------------
```

After formatting a record in DE.O.HEADER is created:

T24

```
Model Bank R14          OUTWARD DELIVERY HEADER SEE

    DATE.TIME.STAMP... D20140516-37540-24206-04
--------------------------------------------------------------------------------
  1 MESSAGE TYPE...... 103
  3 APPLICATION....... FTOT
  4 DISPOSITION....... FORMATTED
  6 PRIORITY.......... N
 10 ACCOUNT NUMBER.... 20656
 11. 1 CUSTOMER NUMBER 100472
 12 CUS. COMPANY...... GB0010001
 13 COMPANY CODE...... GB0010001
 16 LANGUAGE.......... GB
 17 VALUE DATE........ 22 APR 2014
 18 CURRENCY.......... USD
 19 AMOUNT............ 9000.00
 20 DEPARTMENT........ 1
 21 TRANSACTION REF... FT141124H6HC
 23 TEST KEY REQ...... NO
 25. 1 CARRIER ADDR NO SWIFT.1
 26. 1 COPY NUMBER.... 1-100472
 27. 1 FRAME NUMBER... 1
 28. 1 FORMAT......... 1
 29. 1 MSG LANGUAGE... GB
 30. 1 MSG PRIORITY... N
 32. 1 MSG DISPOSITION WACK
 34. 1 SHORT NAME..... BANK OF NEWYOR
 36. 1. 1 TO ADDRESS.. IRVTUS33
 38. 1 FROM ADDRESS... DEMOGBPXAXXX
 40. 1 SENT STAMP..... 20140516 06:43:26
 41. 1 CARRIER SEQ NO. 141120TV8563600
 53 BANK DATE......... 22 APR 2014
 60 CURR.NO........... 1
 61. 1 INPUTTER....... 37540_SUPERVISOR_OFS_SEAT
 62. 1 DATE.TIME...... 16 MAY 14 06:43
 63 AUTHORISER........ 37540_SUPERVISOR_OFS_SEAT
 64 CO.CODE........... GB-001-0001         Model Bank R14
 65 DEPT.CODE......... 1                   Implementation
--------------------------------------------------------------------------------
```

MSG DISPOSITION is “Waiting for acknowledgement”; this is stipulated by ACK.REQUIRED being set to Y in DE.INTERFACE record shown earlier.  We can see the message body in F.DE.O.MSG.SWIFT (attaching the tail “.1” to delivery reference since we’re searching for the information contained in the first multivalue set of fields 25 - 41):

jsh

```
CT F.DE.O.MSG.SWIFT 'D20140516375402420604.1'
```

Output

```
    D20140516375402420604.1
001 {1:F01DEMOGBPXAXXX.SN...ISN.}{2:I103IRVTUS33XXXXN}{3:{108:xxxxx}}{4:..:20:
    FT141124H6HC..:23B:CRED..:32A:140422USD9000,00..:50F:/36293..1/Dell Comput
    er..2/1 DELL WAY..3/US/ROUND ROCK..:59:PAULA GATES..:70:PAYMENT OF INVOICE
     NO.23701..:71A:SHA..:72:/REC/PLEASE EXECUTE THE PAYMENT..-}
```

Also, let’s see the deal that generated this message (its @ID can be found not only in tag 20 of the message above but also in TRANSACTION.REF field of DE.O.HEADER record):

T24

```
 Model Bank R14          FUNDS.TRANSFER SEE                    REF FT141124H6HC


 ------------------------------------------------------------------------------
   1 TRANSACTION.TYPE.. OT03               Outward Swift Payment MT103
   2 DEBIT.ACCT.NO..... 36293              DELLUSDBR2
   4 CURRENCY.MKT.DR... 1                  Currency Market
   5 DEBIT.CURRENCY.... USD                US Dollar
   7 DEBIT.VALUE.DATE.. 22 APR 2014
  11 CREDIT.ACCT.NO.... 20656              BANK OF NEW YORK
  12 CURRENCY.MKT.CR... 1                  Currency Market
  13 CREDIT.CURRENCY... USD                US Dollar
  14 CREDIT.AMOUNT..... 9,000.00
  15 CREDIT.VALUE.DATE. 22 APR 2014
  18 PROCESSING.DATE... 22 APR 2014
  27. 1 BEN.CUSTOMER... Paula Gates
  32. 1 PAYMENT.DETAILS Payment of Invoice no.23701
  41 BEN.OUR.CHARGES... SHA
  45 COMMISSION.CODE... DEBIT PLUS CHARGES
  46. 1 COMMISSION.TYPE OUTREMITCHG        Outward REMIT Charges
 ------------------------------------------------------------------------------
 16 FEB 2015 07:05:12  USER (22 APR) VLADIMIR.K       [28222,IPAGE 1   >>>4>>>
 ACTION _
 AWAITING PAGE INSTRUCTIONS
```

Delivery reference can be seen here:

T24

```
  80. 1 DELIVERY.OUTREF D20140516375402420603-900.1.1      DEBIT ADVICE
  80. 2 DELIVERY.OUTREF D20140516375402420604-103.1.1      CABLE TO CUST
```

We can see that besides MT103 message another one (type 900) was also generated.

Let’s see it in DE.O.HEADER application (go to page 2):

T24

```
Model Bank R14          OUTWARD DELIVERY HEADER SEE

    DATE.TIME.STAMP... D20140516-37540-24206-03
------------------------------------------------------------------------------
 25. 1 CARRIER ADDR NO PRINT.1
 26. 1 COPY NUMBER.... 1-100283
 27. 1 FRAME NUMBER... 1
 28. 1 FORMAT......... 1
 29. 1 MSG LANGUAGE... GB
 30. 1 MSG PRIORITY... N
 32. 1 MSG DISPOSITION FORMATTED
 34. 1 SHORT NAME..... Dell Computer
 36. 1. 1 TO ADDRESS.. Dell Computer
 36. 1. 2 TO ADDRESS.. 1 DELL WAY
 36. 1. 3 TO ADDRESS.. ROUND ROCK
 44. 1 FORM TYPE...... ADVICE
 53 BANK DATE......... 22 APR 2014
 60 CURR.NO........... 1
 61. 1 INPUTTER....... 37540_SUPERVISOR_OFS_SEAT
 62. 1 DATE.TIME...... 16 MAY 14 06:43
------------------------------------------------------------------------------
16 FEB 2015 07:07:37  USER (22 APR) VLADIMIR.K        [28222,IPAGE 1   >>>3>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Carrier here is PRINT (online service BNK/PRINT.OUT is responsible for formatting). However, file F.DE.O.MSG.PRINT is empty.  We can check DE.FORM.TYPE application, record ADVICE (mentioned in the field 44.1):

T24

```
Model Bank R14          DE.FORM.TYPE SEE

    FORM.TYPE......... ADVICE
------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. DELIVERY ADVICE A4 FORMAT
  2 PRINTER.ID........ HOLD                Hold Output - Do Not Print
  3 FORM.WIDTH........ 80
  4 FORM.DEPTH........ 66
  8. 1 OPTIONS........ NOHEAD
  8. 2 OPTIONS........ FORM ADVICE
 10 CURR.NO........... 5
 11. 1 INPUTTER....... 1_201312m
 12. 1 DATE.TIME...... 28 MAR 14 12:50
 13 AUTHORISER........ 98220_INPUTTER_OFS_MB.OFS.AUTH
 14 CO.CODE........... GB-001-0001         Model Bank R14
 15 DEPT.CODE......... 200                 Implementation




------------------------------------------------------------------------------
```

Final check - see field 2 contents referring to application PRINTER.ID:

T24

```
Model Bank R14          PRINTER FILE MAINTENANCE SEE

    PRINTER.NAME...... HOLD
--------------------------------------------------------------------------------
  1 PRIME.PRINTER.ID.. HOLD
  2 DESCRIPTION....... Hold Output - Do Not Print
  3 PRINTER.TYPE......
  4. 1. 1 COMMAND.....
  5 RESERVE1..........
  6 RESERVE2..........
  7 RESERVE3..........
  8 RESERVE4..........
  9 RESERVE5..........
 10 RESERVE6..........
 11 RECORD.STATUS.....
 12 CURR.NO........... 2
 13. 1 INPUTTER....... 1_G10.1.02m
 14. 1 DATE.TIME...... 29 OCT 99 14:21
 15 AUTHORISER........ 16_INPUTTER
 16 CO.CODE........... GB-001-0001         Model Bank R14
--------------------------------------------------------------------------------
16 FEB 2015 07:28:36  USER (22 APR) VLADIMIR.K        [9072,INPAGE 1   >>>2>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

In the field 4 - COMMAND - one or several commands or programs can be specified to perform some action when an advice is “printed” - e.g.  to save the output elsewhere.  Example of that is given later in this book.

Customer addresses are stored in DE.ADDRESS. See addresses used in the messages shown above:

T24 DE.ADDRESS

```
    ID................ GB0010001.C-100472.SWIFT.1
--------------------------------------------------------------------------------
  1 DELIVERY ADDRESS.. IRVTUS33
  6. 1 GB SHORT NAME.. BANK OF NEWYOR
 31 CURR.NO........... 1
 32. 1 INPUTTER....... 91944_OFFICER__OFS_SEAT
 33. 1 DATE.TIME...... 08 APR 14 17:12
 34 AUTHORISER........ 91944_OFFICER_OFS_SEAT
 35 CO.CODE........... GB-001-0001         Model Bank R14
 36 DEPT.CODE......... 1                   Implementation
```

T24 DE.ADDRESS

```
    ID.............. GB0010001.C-100283.PRINT.1
--------------------------------------------------------------------------------
  6. 1 GB SHORT NAME.. Dell Computer
  7. 1 GB NAME 1...... Dell Computer
  9. 1 GB STREET ADDR. 1 DELL WAY
 10. 1 GB TOWN COUNTRY ROUND ROCK
 22 ADDR.LOCATION..... PRIMARY
 31 CURR.NO........... 1
 32. 1 INPUTTER....... 73318_OFFICER__OFS_SEAT
 33. 1 DATE.TIME...... 08 APR 14 17:09
 34 AUTHORISER........ 73318_OFFICER
 35 CO.CODE........... GB-001-0001         Model Bank R14
 36 DEPT.CODE......... 1                   Implementation
```

*(DE.ADDRESS record with the suffix “PRINT.1” is created automatically upon authorisation of new CUSTOMER record.)*

If message formatting resulted in an error (e.g.  customer address missing), message is marked as “REPAIR”. If the error occured at the mapping stage - field DISPOSITION is populated accordingly:

T24

```
Model Bank R14          OUTWARD DELIVERY HEADER INPUT

    DATE.TIME.STAMP... D20140515-15686-82218-19
--------------------------------------------------------------------------------
  1 MESSAGE TYPE...... 752
  2 APPLICATION FORMAT
  3 APPLICATION....... LC
  4 DISPOSITION....... REPAIR
  5. 1 ERROR CODE..... ERROR - 752.LC.4-Mandatory message fields (&) missing#FURTHER.
  6 PRIORITY.......... N
  7 STATUS............
  8 DESTINATION CODE..
  9 POS DUP ENTRY.....
 10 ACCOUNT NUMBER....
 11. 1 CUSTOMER NUMBER
 12 CUS. COMPANY......
 13 COMPANY CODE......
 14. 1 OVR DEL ADD....
 15 OVERRIDE CARRIER..
 16 LANGUAGE..........
--------------------------------------------------------------------------------
16 FEB 2015 07:47:25  USER (22 APR) VLADIMIR.K        [9072,INPAGE 1   >>>5>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Error at formatting stage results in MSG.DISPOSITION being set:

T24

```
Model Bank R14          OUTWARD DELIVERY HEADER INPUT

    DATE.TIME.STAMP... D20140515-12424-79292-04
--------------------------------------------------------------------------------
  1 MESSAGE TYPE...... 769
  2 APPLICATION FORMAT
  3 APPLICATION....... MD
  4 DISPOSITION....... UNFORMATTED
  5. 1 ERROR CODE.....
  6 PRIORITY.......... N
  7 STATUS............
  8 DESTINATION CODE..
  9 POS DUP ENTRY.....
 10 ACCOUNT NUMBER....
 11. 1 CUSTOMER NUMBER 100282
 12 CUS. COMPANY...... GB0010001
 13 COMPANY CODE...... GB0010001
 14. 1 OVR DEL ADD....
 15 OVERRIDE CARRIER..
 16 LANGUAGE.......... GB
 17 VALUE DATE........
 18 CURRENCY..........
 19 AMOUNT............
 20 DEPARTMENT........ 1
 21 TRANSACTION REF... MD1408364080
 22 APPLICATION QUEUE.
 23 TEST KEY REQ...... NO
 24 INWARD PAY TYPE...
 25. 1 CARRIER ADDR NO SWIFT.1
 26. 1 COPY NUMBER.... 1-100282
 27. 1 FRAME NUMBER... 1
 28. 1 FORMAT......... 1
 29. 1 MSG LANGUAGE... GB
 30. 1 MSG PRIORITY... N
 31. 1 MSG STATUS.....
 32. 1 MSG DISPOSITION REPAIR
 33. 1. 1 MSG ERR CDE. ERROR - Carrier Address SWIFT does not exist for the customer
--------------------------------------------------------------------------------
```

If the setup error is corrected, such messages can be processed by putting RESUBMIT into corresponding field.

Another application of interest is DE.DISP.CONTROL. It allows to assign a status to a message according to some logic.

In the following example to all Forex-related messages MT202 will be applied the status “HOLD”:

T24

```
Model Bank R14          DISPOSITION CONTROL INPUT

    STATUS.KEY........ 1
--------------------------------------------------------------------------------
  1. 1 FIELD.NAME..... APPLICATION
  2. 1 OPERAND........ EQUAL
  3. 1 CONDITION...... FX
  1. 2 FIELD.NAME..... MESSAGE.TYPE
  2. 2 OPERAND........ EQUAL
  3. 2 CONDITION...... 202
  1. 3 FIELD.NAME..... STATUS
  2. 3 OPERAND........ EQUAL
  3. 3 CONDITION...... HOLD
  4. 1 FIELD.POS...... 3
  4. 2 FIELD.POS...... 1
  4. 3 FIELD.POS...... M31
  5 STATUS............
  6 PRIORITY.......... N
  7 HOLD.UNTIL.DATE...
  8 RECORD.STATUS.....
--------------------------------------------------------------------------------
16 FEB 2015 08:00:05  USER (22 APR) VLADIMIR.K        [9072,INPAGE 1   >>>2>>>
ACTION
AWAITING PAGE INSTRUCTIONS
```

*(HOLD status of delivery message has nothing in common neither with HOLD status of application record nor with &HOLD& file where T24 printouts are stored.)*

*(Status can also be assigned to “DELETED” to totally get rid of the particular message types.)*

*(To enable this functionality field DISP.CONTROL of record SYSTEM in application DE.PARM has to be set to Y.)*

To the field FIELD.NAME a hook routine can be attached for more complex analysis, e.g.:

T24

```
  1. 3 FIELD.NAME..... @TEST.FX.DATA.
  2. 3 OPERAND........ EQUAL
  3. 3 CONDITION...... 1
  1. 4 FIELD.NAME..... STATUS
  2. 4 OPERAND........ EQUAL
  3. 4 CONDITION...... HOLD
```

Parameters are:  HEADER.REC, OPERAND, CONDITION, RETURN.FLAG.

HEADER.REC is DE.O.HEADER record, deal record (R.NEW) is also available; RETURN.FLAG is to be set to 1 in this example to apply the desired status.

To manually release a message that is in HOLD status set MSG STATUS to RELEASE in its DE.O.HEADER record.

**Inward delivery**.

Online service BNK/SWIFT.IN parses the incoming messsages and firstly puts them to F.DE.I.MSG table; example:

jsh

```
CT F.DE.I.MSG 'R20140516388702481901'
```

Output

```
    R20140516388702481901
001 :20:FT0711000333..:23B:CRED..:32A:140422GBP1000,00..:33B:GBP1000,00..:50K:
    E.D + F MAN SUGAR LTD..SUGAR QUAY LOWER THAMES ST..LONDON EC3R 6DU..:53B:/
    D/BARC-238487488388383..:59:/10995..COCACOLA INDUSTRIES..PARK AVENUE..LOND
    ON SQUARE..LONDON..:70:PAYMENT FOR INV 141..:71A:SHA..-}
```

*(Inward delivery reference starts with “R” rather than from “D” as outward does.)*

Record in DE.I.HEADER reflects creation of FT via OFS:

T24

```
 Model Bank R14          INWARD DELIVERY HEADER SEE

     DATE.TIME.STAMP... R20140516-38870-24819-01
 ------------------------------------------------------------------------------
   1 MESSAGE TYPE...... 103
   4 DISPOSITION....... OFS FORMATTED
   6 PRIORITY.......... N
  11 CUSTOMER NUMBER... 100461
  12 CUS. COMPANY...... GB0010001
  13 COMPANY CODE...... GB0010001
  17 VALUE DATE........ 22 APR 2014
  18 CURRENCY.......... GBP
  19 AMOUNT............ 1000.00
  21 TRANSACTION REF... FT0711000333
  25. 1 CARRIER ADDR NO SWIFT
  36. 1 TO ADDRESS..... DEMOGBPXAXXX
  37. 1 RECEIVED STAMP. 06:53:39
  38. 1 FROM ADDRESS... BARCGB22XXXX
  40. 1 SENT STAMP..... 06:53:39
  41. 1 CARRIER SEQ NO. SCRPT09180000101-01-I
  53 BANK DATE......... 22 APR 2014
  55. 1 INW. {1:F01DEMOGBPXAXXX.SN...ISN.}
  55. 2 INW. {2:I103BARCGB22XXXXN}
  55. 3 INW. {3:{108:xxxxx}}
  55. 4 INW. {5:
  56. 1 OFS. INW141367049424820.00
  57. 1 T24. FT141127WPJ0
  60 CURR.NO........... 1
  61. 1 INPUTTER....... 1_SYSTEM
  62. 1 DATE.TIME...... 16 MAY 14 06:53
  63 AUTHORISER........ 1_SYSTEM
  64 CO.CODE........... GB-001-0001         Model Bank R14
  65 DEPT.CODE......... 1                   Implementation
 ------------------------------------------------------------------------------
```

See the deal (it’s not FT0711000333 mentioned in the field 21 - that’s sender reference which apparently also uses T24 but doesn’t use unique FT @ID generation). Record is large; only some fields are shown at the following screenshot:

T24

```
Model Bank R14          FUNDS.TRANSFER SEE                    REF FT141127WPJ0
TELEX FROM  Barclays London    MT103

--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE.. IT                  Inward SWIFT Payment
  ...
  9 DEBIT.THEIR.REF... FT0711000333
  ...
 19. 1 ORDERING.CUST.. E.D + F MAN SUGAR LTD
 19. 2 ORDERING.CUST.. SUGAR QUAY LOWER THAMES ST
 19. 3 ORDERING.CUST.. LONDON EC3R 6DU
 20. 1 IN.ORDERING.CUS E.D + F MAN SUGAR LTD
 20. 2 IN.ORDERING.CUS SUGAR QUAY LOWER THAMES ST
 20. 3 IN.ORDERING.CUS LONDON EC3R 6DU
 21. 1 ORDERING.BANK.. 100461              Barclays London
 22. 1 IN.ORDERING.BK. 100461
 26 IN.BEN.ACCT.NO.... 10995
 28. 1 IN.BEN.CUSTOMER COCACOLA INDUSTRIES
 28. 2 IN.BEN.CUSTOMER PARK AVENUE
 28. 3 IN.BEN.CUSTOMER LONDON SQUARE
 28. 4 IN.BEN.CUSTOMER LONDON
 32. 1 PAYMENT.DETAILS PAYMENT FOR INV 141
  ...
 70 AMOUNT.DEBITED.... GBP 1,000.00
 71 AMOUNT.CREDITED... USD 960.00
 74 CUSTOMER.RATE..... 0.96
 76 INWARD. MT103
 78 TELEX.FROM.CUST... Barclays London
 79 DELIVERY.INREF.... R20140516388702481901
  ...
125. 1 IN.P FIELD NOT MAPPED FOR TAG -23B
126. 1 IN.S :20:FT0711000333
126. 2 IN.S :23B:CRED
126. 3 IN.S :32A:140422GBP1000,00
126. 4 IN.S :33B:GBP1000,00
126. 5 IN.S :50K:E.D + F MAN SUGAR LTD
126. 6 IN.S SUGAR QUAY LOWER THAMES ST
126. 7 IN.S LONDON EC3R 6DU
  ...
126. 8 IN.S :53B:/D/BARC-238487488388383
126. 9 IN.S :59:/10995
126.10 IN.S COCACOLA INDUSTRIES
126.11 IN.S PARK AVENUE
126.12 IN.S LONDON SQUARE
126.13 IN.S LONDON
126.14 IN.S :70:PAYMENT FOR INV 141
126.15 IN.S :71A:SHA
  ...
144 INW.SEND.BIC...... BARCGB22XXXX
  ...
197 RECORD.STATUS..... INAU              INPUT Unauthorised
199. 1 INPUTTER....... 18_INPUTTER__OFS_SWIFTIN
```

Inward delivery can be customized; routines to process message tags are declared at DE.I.SUBROUTINE.TABLE application, e.g.  for SWIFT tag 32:

T24

```
Model Bank R14          INWARD DELIVERY SUBROUTINES INPUT

    TAG.KEY........... 32
--------------------------------------------------------------------------------
  1 SUBROUTINE........ DE.I.TAG32          INWARD TAG ROUTINE
```

A local subroutine can be attached here; the template for it can be found in T24.BP:

jsh

```
CT T24.BP DE.I.TAG32
```

Output

```
001     SUBROUTINE DE.I.TAG32(TAG,DATA32,OFS.DATA,SPARE1,SPARE2,SPARE3,SPARE4,
    DE.I.FIELD.DATA,TAG.ERR)
002 *-----------------------------------------------------------------------------
    ----
003 * <Rating>-36</Rating>
004 ************************************************************************
    **********************
005 *
006 * This routine assigns SWIFT tag32 - Date , Currency , Amount to the ofs m
    essage being
007 * build up via inward delivery
008 *
009 * Inward
010 *  Tag            -  The swift tag 32A , 32B
011 *  Data32         -  The swift data
012 *
013 * Outward
014 *  OFS.DATA       - The corresponding application field in OFS format
015 ************************************************************************
    **********
...
055     BEGIN CASE
056     CASE TAG = '32A'
057 * BG_100007343 - S
058
059         DDATE = DATA32[1,6]
060         YY = DATA32[1,2]
061         MM = DATA32[3,2]
062         DD = DATA32[5,2]
063         OFS.DATA := DATE.FIELD :'=':DD:MM:YY
...
```

*(Source code for subroutine DE.I.MT103 that processes the message itself can also be found in T24.BP but it’s not necessarily the latest one.)*

**BIC directory**.

There is correspondent T24 application:

T24

```
 Model Bank R14


 --------------------------------------------------------------------------------
 ABNRNLNL LIMBURG   ABN AMRO BANK
 AEIBCHGV           AMERICAN EXPRESS BAN
                    SWITZERLAND
 AEIBFRPP PARIS     AMERICAN EXPRESS BAN
                    FRANCE
 AEIBGB22           AMERICAN EXPRESS BAN
 AEIBLUBR           AMERICAN EXPRESS BAN
                    LUXEMBURG
 AEIBSGSG           AMERICAN EXPRESS BAN
 AEIBUS33           AMERICAN EXPRESS BAN
 AFRINGPP
 AIBKGB22           ALLIED IRISH BAN
1AIBKIEDB           ALLIED IRISH BAN
1AIBKSGSG           ALIIED IRISH BAN
1ALRASAPP
1ANZBAUML           ANZ GRINDLAYS BAN
 -------------------AUSTRALIA                        ------------------------
1BARCGB22           BARCLAYS BANK PL                 [32400,IPAGE   1 >>>2>>>
1BBMIMYKL           BANK BUMIPUTR
7AWAITING PAGE INSTRUCTIONS
```

T24

```
 Model Bank R14          S.W.I.F.T. BIC Database SEE

     BICID.............. AIBKGB22
 --------------------------------------------------------------------------------
   4. 1 INSTITUTION.... ALLIED IRISH BAN
   7 BIC.CODE.......... AIBKGB22
  16 COUNTRY.CODE...... GB
  32. 1 COUNTRY........ GREAT BRITAIN
  41 SWIFT.AUTH.KEY.... YES
  53 MANUAL............ Y
  66 CURR.NO........... 1
  67. 1 INPUTTER....... 50312_OFFICER__OFS_SEAT
  68. 1 DATE.TIME...... 08 APR 14 17:01
  69 AUTHORISER........ 50312_OFFICER_OFS_SEAT
  70 CO.CODE........... GB-001-0001         Model Bank R14
  71 DEPT.CODE......... 1                   Implementation




 --------------------------------------------------------------------------------
 18 FEB 2015 05:53:02  USER (22 APR) VLADIMIR.K        [32400,IPAGE 1
 ACTION _
 AWAITING PAGE INSTRUCTIONS
```

These records are created from BIC directory file using application DE.BIC.LOAD:

T24

```
Model Bank R14          S.W.I.F.T BIC Database Upload INPUT

    DE.BIC.LOAD....... SYSTEM
------------------------------------------------------------------------------
  1. 1 DESCRIPTION.... _
  2 RESERVED.11.......
  3 FILE.NAME.........
  4 FILE.LOCATION.....
  5 DELIMITER.........
  6 RESERVED10........
  7 RESERVED9.........
  8 RESERVED8.........
  9 RESERVED7.........
 10 RESERVED6.........
 11 RESERVED5.........
 12 RESERVED4.........
 13 RESERVED3.........
 14 RESERVED2.........
 15 RESERVED1.........
 16. 1 LOCAL.REF......
------------------------------------------------------------------------------
18 FEB 2015 05:54:22  USER (22 APR) VLADIMIR.K        [32400,IPAGE 1   >>>2>>>
ACTION
```

## 1.77 NOFILE enquiry

The “regular” enquiry isn’t able to gather data from more than one file. “NOFILE” one can do that. For example, we want to display record @IDs from both LIVE and NAU file of FUNDS.TRANSFER.

Step 1.  Routine:

jsh

```
SUBROUTINE NOFILE.ALL.FT(ret_list)
* T24 nofile enquiry routine
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_ENQUIRY.COMMON

   HUSH ON
   EXECUTE 'SSELECT FBNK.FUNDS.TRANSFER' RTNLIST live_list
   EXECUTE 'SSELECT FBNK.FUNDS.TRANSFER$NAU' RTNLIST nau_list
   HUSH OFF

   len_live = DCOUNT(live_list, FM)
   len_nau = DCOUNT(nau_list, FM)

   FOR i = 1 TO len_live
      ret_list<-1> = 'L*' : live_list<i>
   NEXT i

   FOR i = 1 TO len_live
      ret_list<-1> = 'N*' : nau_list<i>
   NEXT i

   RETURN
END
```

Step 2.  Create STANDARD.SELECTION record:

T24

```
 Model Bank R14          STANDARD SELECTION FIELDS, SEE

     FILE.NAME......... NOFILE.ALL.FT.ENQ
 ------------------------------------------------------------------------------
  15. 1 USR.FIELD.NAME. RTNDATA
  16. 1 USR.TYPE....... R
  17. 1. 1 USR.FIELD.NO NOFILE.ALL.FT                    <===== routine name here
  20. 1 USR.DISPLAY.FMT 64L
  24. 1 USR.SINGLE.MULT S
  15. 2 USR.FIELD.NAME. RAWDATA
  16. 2 USR.TYPE....... D
  17. 2. 1 USR.FIELD.NO 0
  20. 2 USR.DISPLAY.FMT 64L
  24. 2 USR.SINGLE.MULT S
  35 CURR.NO........... 1
  36. 1 INPUTTER....... 32026_VLADIMIR.K
  37. 1 DATE.TIME...... 17 FEB 15 07:07
  38 AUTHORISER........ 32026_VLADIMIR.K
  39 CO.CODE........... GB-001-0001         Model Bank R14
  40 DEPT.CODE......... 1                   Implementation
 ------------------------------------------------------------------------------
 17 FEB 2015 07:07:19  USER (22 APR) VLADIMIR.K        [32026, PAGE 1
 ACTION _
 AWAITING PAGE INSTRUCTIONS
```

Step 3.  Create ENQUIRY:

T24

```
Model Bank R14          ENQUIRY SEE

    ENQUIRY........... ALL.FT
------------------------------------------------------------------------------
  1 PAGE.SIZE ........ 4,19
  2 FILE.NAME......... NOFILE.ALL.FT.ENQ                       <===== SS record @ID
  3. 1 FIXE RTNDATA LK ...    <===== FIXED.SELECTION, otherwise routine wouldn't run
 14. 1 FIELD.NAME..... RAWDATA                                 <===== SS field (15.2)
 15. 1. 1 OPERATION... RAWDATA
 16. 1 COLUMN......... 4
 17. 1 LENGTH.MASK.... 4L
 18. 1. 1 C F *,1                          <===== CONVERSION; take data before "*"
 18. 1. 2 C SUBSTITUTE L,LIVE              <===== then output LIVE instead of L
 18. 1. 3 C SUBSTITUTE N,$NAU              <=====   or $NAU instead of N
 20. 1 TYPE........... L
 14. 2 FIELD.NAME..... RECID
 15. 2. 1 OPERATION... RAWDATA
 16. 2 COLUMN......... 9
 17. 2 LENGTH.MASK.... 64L
 18. 2. 1 C F *,2                           <===== CONVERSION; take data after "*"
 20. 2 TYPE........... L
 40 PAGE.FIELDS....... 2 1?2
 42 MULTI.FIELDS...... 0
 43 BREAK.FIELDS...... 0
 44 PROCESS.BREAKS.... 0
 45 TOTAL.FIELDS...... 0
 77 CURR.NO........... 4
 78. 1 INPUTTER....... 6732_VLADIMIR.K
 79. 1 DATE.TIME...... 17 FEB 15 07:18
 80 AUTHORISER........ 6732_VLADIMIR.K
 81 CO.CODE........... GB-001-0001         Model Bank R14
 82 DEPT.CODE......... 1                   Implementation
------------------------------------------------------------------------------
```

To be able to see the particular record - add the following setting to enquiry:

T24

```
 36. 1 ENQU FUNDS.TRANSFER S RECID
 37. 1. 1 S
 38. 1 LABEL.FIELD.... RECID 2
 39. 1. 1 GB NXT.DESC. Go to record
```

Run enquiry:

T24

```
Model Bank R14


--------------------------------------------------------------------------------
1  LIVE FT14098HFGC0
2  LIVE FT1409868VHT
3  LIVE FT14112Q080W
4  LIVE FT14112SXZSX
5  LIVE FT14112VCB74
6  LIVE FT14112TLKB1
7  LIVE FT14112PRFVK
8  LIVE FT141120PWRV
9  LIVE FT140989HFGX
10  LIVE FT14098L0ZCW
11  LIVE FT141124QMTT
12  LIVE FT14112PDKRF
13  LIVE FT14112YJ9B6
14  LIVE FT1411213JX8
15  LIVE FT14112R8226
16  LIVE FT14112XJPW9
--------------------------------------------------------------------------------
17 FEB 2015 07:26:54  USER (22 APR) VLADIMIR.K        [6732,INPAGE   1 >>>3>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Scroll down to where NAU records begin:

T24

```
Model Bank R14


--------------------------------------------------------------------------------
1  LIVE FT14112JGB9Y
2  LIVE FT14112K687M
3  LIVE FT14098W9XZT
4  LIVE FT1409861K0J
5  LIVE FT141124H6HC
6  LIVE FT14098N6MGH
7  LIVE FT14098M2KLV
8  $NAU FT141126TL15
9  $NAU FT14107M8DJG
10  $NAU FT14112H3HM6
11  $NAU FT14104P990L
12  $NAU FT14097Z8FQC
13  $NAU FT1411234Z04
14  $NAU FT140870BGYX
15  $NAU FT14093Z83RK
16  $NAU FT1410788Q4X
--------------------------------------------------------------------------------
17 FEB 2015 07:26:54  USER (22 APR) VLADIMIR.K        [6732,INPAGE   7 >>>9>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

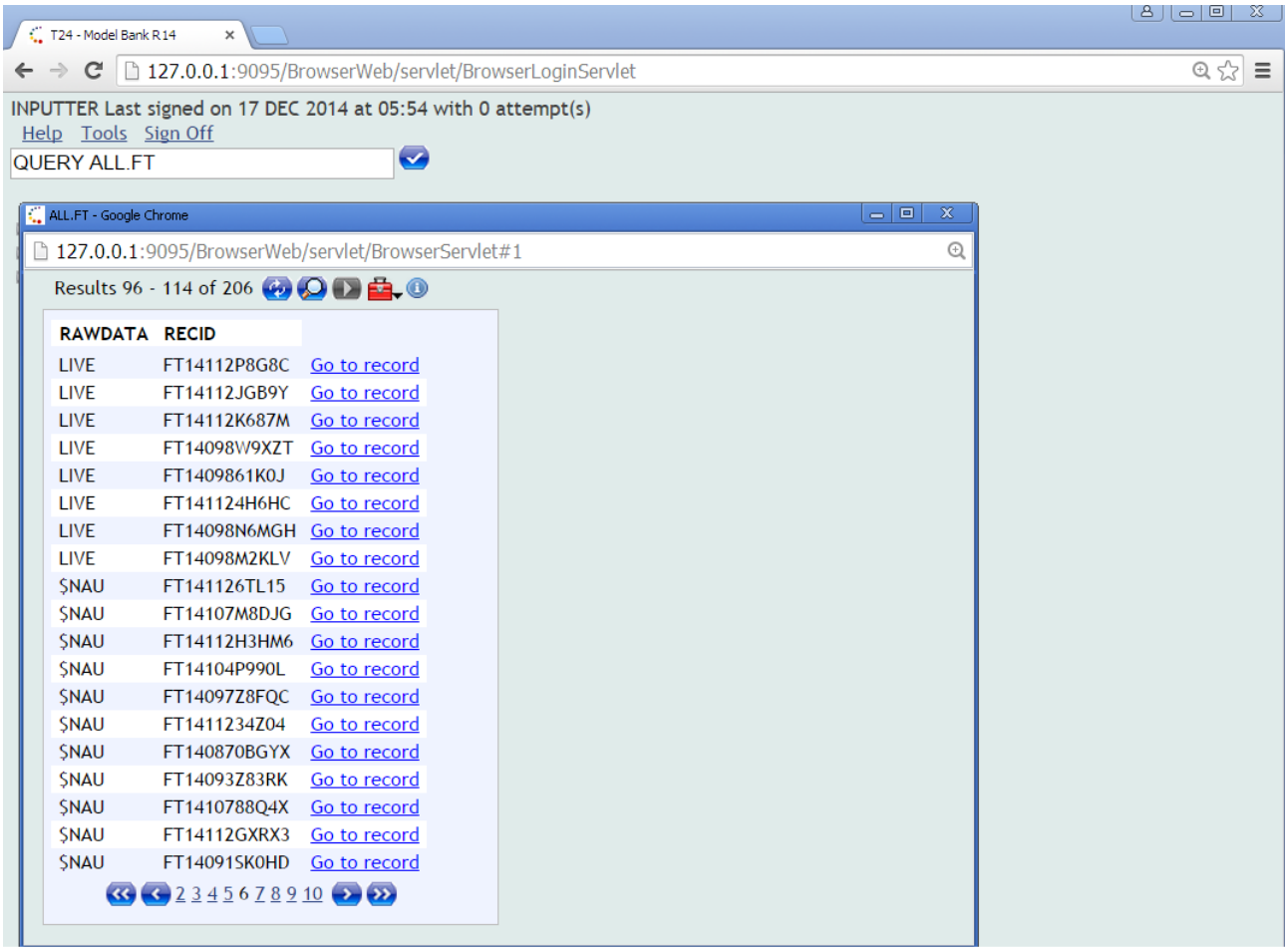


**Figure 44: Same place in Browser.**

## 1.78 Menus

HEPTEXT.MENU application is used to create a menu, HELPTEXT.MAINMENU is used to group menus together.

A menu can be invoked by itself or by choosing it in a group.

T24

```
Model Bank R14          HELPTEXT.MENU, SEE

    MENU.NAME......... MY.ADMIN.MENU
--------------------------------------------------------------------------------
  1. 1 APPL SPF S SYSTEM
  2. 1. 1 GB DESCRIPT. See spf record
  1. 2 APPL COMPANY L
  2. 2. 1 GB DESCRIPT. Branches list
  1. 3 APPL VERSION E
  2. 3. 1 GB DESCRIPT. Unauthorised VERSIONs
  1. 4 APPL ENQUIRY E
  2. 4. 1 GB DESCRIPT. Unauthorised ENQUIRY's
  3 DISPL.APPLICATION. Y
  6 CURR.NO........... 2
  7. 1 INPUTTER....... 10898_VLADIMIR.K
  8. 1 DATE.TIME...... 18 FEB 15 06:35
  9 AUTHORISER........ 10898_VLADIMIR.K
 10 CO.CODE........... GB-001-0001         Model Bank R14
 11 DEPT.CODE......... 1                   Implementation

--------------------------------------------------------------------------------
18 FEB 2015 06:35:23  USER (22 APR) VLADIMIR.K        [10898, PAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Invoke the menu - command ?MY.ADMIN.MENU at “AWAITING APPLICATION” prompt:

T24

```
Model Bank R14         MENU MY.ADMIN.MENU

NO. DESCRIPTION                                 APPLICATION NAME
--------------------------------------------------------------------------------
  1 See spf record                              SPF S SYSTEM
  2 Branches list                               COMPANY L
  3 Unauthorised VERSIONs                       VERSION E
  4 Unauthorised ENQUIRY's                      ENQUIRY E









--------------------------------------------------------------------------------
18 FEB 2015 06:37:41  USER (22 APR) VLADIMIR.K        [10898,IPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Same command works in Browser.

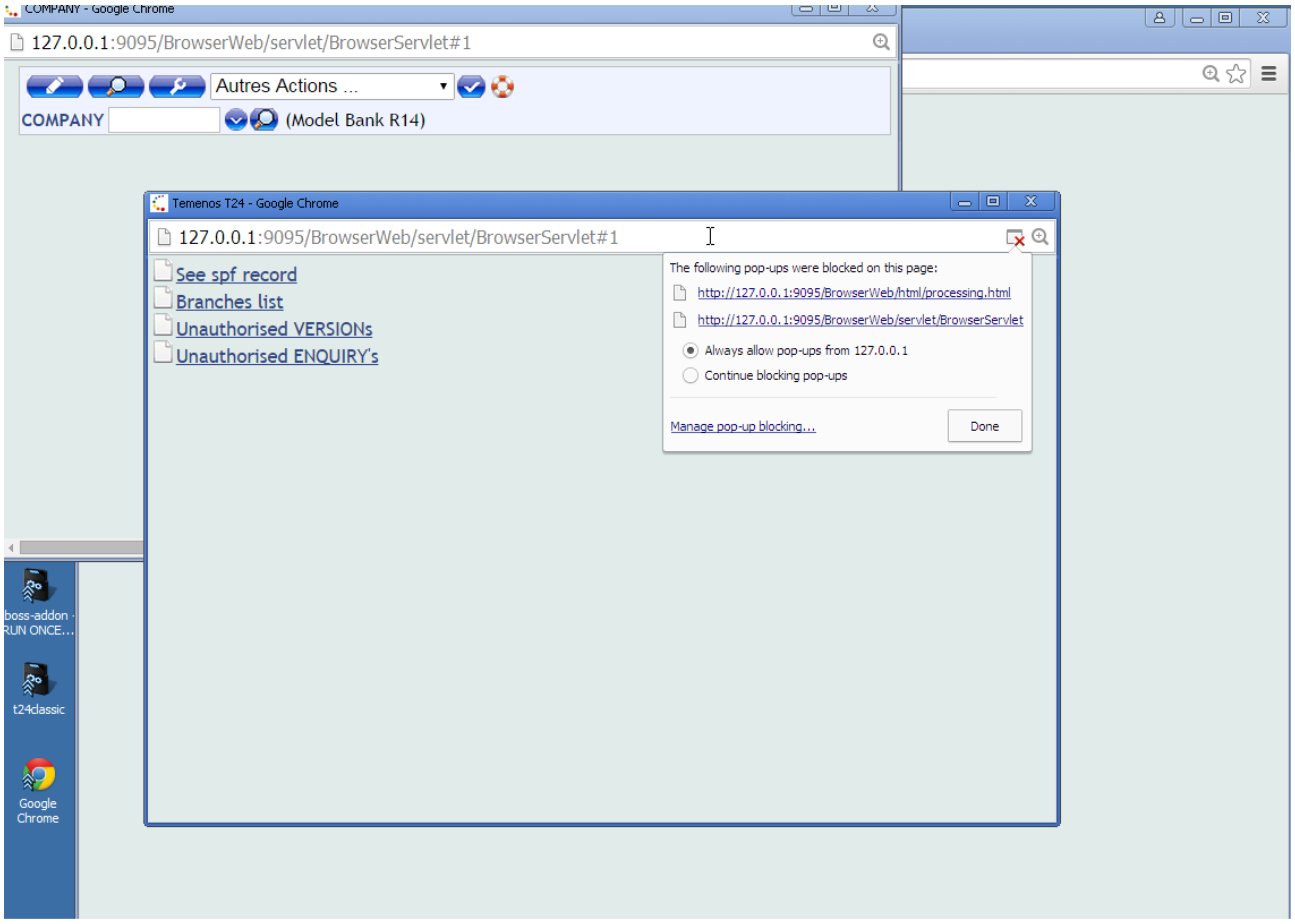


**Figure 45:  Menu invocation in Browser.**

*(If after click to "Branches list" it doesn't appear, allow pop-ups from the application server, as it's shown at the screenshot.)*

Now we'll create the second menu and link them together.  Step 1:

T24

```
Model Bank R14          HELPTEXT.MENU, SEE

    MENU.NAME......... MY.USER.MENU
-------------------------------------------------------------------------------
  1. 1 APPL FT I F3
  2. 1. 1 GB DESCRIPT. New money transfer
  1. 2 APPL TELLER I F3
  2. 2. 1 GB DESCRIPT. New cash transaction
  1. 3 APPL FT E
  2. 3. 1 GB DESCRIPT. Money transfer - unauthorised
  1. 4 APPL TELLER E
  2. 4. 1 GB DESCRIPT. Cash txns - unauthorised
  3 DISPL.APPLICATION. Y
-------------------------------------------------------------------------------
```

Step 2:

T24

```
Model Bank R14          HELPTEXT.MAINMENU, SEE

    ACTION.CODE....... 801
------------------------------------------------------------------------------
  1. 1 GB TITLE....... MY MENU
  2. 1 ID.OF.HELP.MENU MY.ADMIN.MENU
  3. 1. 1 GB DESCRIPT. Administration
  2. 2 ID.OF.HELP.MENU MY.USER.MENU
  3. 2. 1 GB DESCRIPT. Financial transactions
  4 DISPLAY.ID.OF.MENU Y
  6 CURR.NO........... 3
  7. 1 INPUTTER....... 10898_VLADIMIR.K
  8. 1 DATE.TIME...... 18 FEB 15 07:03
  9 AUTHORISER........ 10898_VLADIMIR.K
 10 CO.CODE........... GB-001-0001         Model Bank R14
 11 DEPT.CODE......... 1                   Implementation




------------------------------------------------------------------------------
18 FEB 2015 07:00:26  USER (22 APR) VLADIMIR.K        [10898, PAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

This menu also can be invoked both in Classic and Browser using command ?801.

Step 3 (optional).  Make this menu to appear automatically upon login:

T24

```
Model Bank R14          USER PROFILE, INPUT

    USER.ID........... VLADIMIR.K1
------------------------------------------------------------------------------
 11. 1 END.TIME....... 24:00
 12 TIME.OUT.MINUTES.. 999
 13 ATTEMPTS.......... 9
 14 INIT.APPLICATION.. ?801                                    <===== here
 15. 1 CUSTOMER.......
 16. 1. 1 ACCOUNT.....
 17. 1 COMPANY.RESTR.. ALL
 18. 1 APPLICATION.... ALL.PG
 19. 1 VERSION........
 20. 1 FUNCTION....... A 2 B C D E F H I L P R S V
 21. 1 FIEL
 22. 1 DATA.COMPARISON
 23. 1 DATA
 24. 1 DATA
 25 SIGN.ON.OFF.LOG... NO
 26 SECURITY.MGMT.L... NO
------------------------------------------------------------------------------
18 FEB 2015 07:07:00  USER (22 APR) VLADIMIR.K        [10898,IPAGE 2   >>>6>>>
ACTION
```

Now see the effect (screenshot taken after manual expanding of the top level):

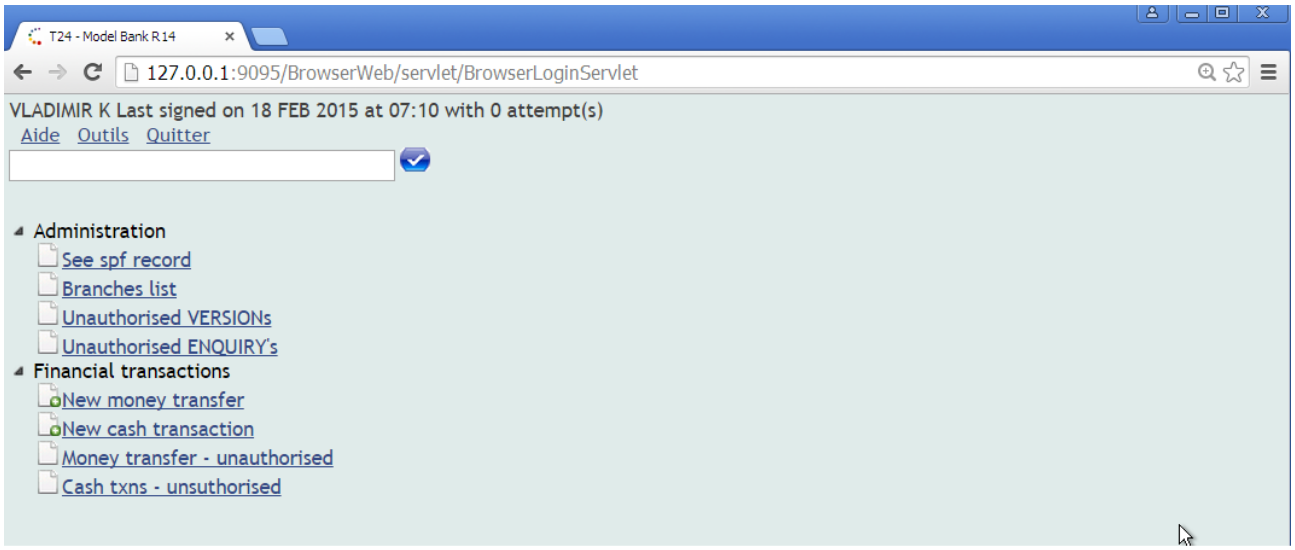


**Figure 46: Menu attached to user profile.**

If we log in with this user using Classic interface, there would be no way to exit T24 since the “AWAITING APPLICATION” level is no more accessible - pressing F1 makes no effect:

T24

```
Model Bank R14          MAINMENU 801
                        MY MENU
NO. DESCRIPTION                                        NAME OF MENU GROUP
------------------------------------------------------------------------------
  1 Administration                                     MY.ADMIN.MENU
  2 Financial transactions                             MY.USER.MENU









-- LAST SIGN.ON, DATE: 18 FEB 2015     TIME: 07:11        ATTEMPTS: 0 --------
18 FEB 2015 07:16:26  USER (22 APR) VLADIMIR.K1       [6654,INPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Any input except menu number results in “EB.RTN.INVALID.INP.2” error.

*(Note that in Browser screenshot above command line is accessible; this is stipulated by USER attribute SUPER.USER.)*

Ctrl-C, however, brings us to debugger prompt and debugger command Q takes us out:

T24 -> jsh

```
 ------------------------------------------------------------------------------
 18 FEB 2015 07:16:35  USER (22 APR) VLADIMIR.K1       [6654,INPAGE 1
 ACTION Interrupt signal                                   EB.RTN.INVALID.INP.2
Line 126 , Source I_SF.INPUT
jBASE debugger->Q
Are you sure ?Y
jBASE debugger , QUIT
jsh r14  -->
```

To avoid this we can create another menu with SIGN.ON command and attach it to our main menu:

T24

```
Model Bank R14          HELPTEXT.MENU, SEE

    MENU.NAME......... EXIT
------------------------------------------------------------------------------
  1. 1 APPL SIGN.ON
  2. 1. 1 GB DESCRIPT. Change user or exit
  3 DISPL.APPLICATION. Y
  6 CURR.NO........... 1
  7. 1 INPUTTER....... 26067_VLADIMIR.K
  8. 1 DATE.TIME...... 18 FEB 15 07:22
  9 AUTHORISER........ 26067_VLADIMIR.K
 10 CO.CODE........... GB-001-0001         Model Bank R14
 11 DEPT.CODE......... 1                   Implementation
```

T24

```
Model Bank R14          HELPTEXT.MAINMENU, SEE

    ACTION.CODE....... 801
------------------------------------------------------------------------------
  1. 1 GB TITLE....... MY MENU
  2. 1 ID.OF.HELP.MENU MY.ADMIN.MENU
  3. 1. 1 GB DESCRIPT. Administration
  2. 2 ID.OF.HELP.MENU MY.USER.MENU
  3. 2. 1 GB DESCRIPT. Financial transactions
  2. 3 ID.OF.HELP.MENU EXIT
  3. 3. 1 GB DESCRIPT. Exit
  4 DISPLAY.ID.OF.MENU Y
  6 CURR.NO........... 4
  7. 1 INPUTTER....... 26067_VLADIMIR.K
  8. 1 DATE.TIME...... 18 FEB 15 07:22
  9 AUTHORISER........ 26067_VLADIMIR.K
 10 CO.CODE........... GB-001-0001         Model Bank R14
 11 DEPT.CODE......... 1                   Implementation
```

Now log in:

T24

```
Model Bank R14         MAINMENU 801
                       MY MENU
NO. DESCRIPTION                                    NAME OF MENU GROUP
------------------------------------------------------------------------------
  1 Administration                                 MY.ADMIN.MENU
  2 Financial transactions                         MY.USER.MENU
  3 Exit                                           EXIT









-- LAST SIGN.ON, DATE: 18 FEB 2015     TIME: 07:16        ATTEMPTS: 0 --------
18 FEB 2015 07:23:35  USER (22 APR) VLADIMIR.K1       [5817,INPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Enter option 3, then 1; on “PLEASE ENTER YOUR SIGN ON NAME” prompt either type new user login name or BK to exit to jsh.

## 1.79 OFS.POST.MESSAGE

This T24 core subroutine in conjunction with online service OFS.MESSAGE.SERVICE can be used for asynchronous processing of OFS messages. It can be called from local code with the following 4 parameters:

- OFS message.
- (return parameter) @ID that will be assigned to it.
- OFS.SOURCE @ID.
- Parameter 4 can be left blank.

After the call OFS message is placed to F.OFS.MESSAGE.QUEUE table; OFS.MESSAGE.SERVICE takes it from there, processes and puts the result to the table F.OFS.RESPONSE.QUEUE.

Example:

jBC

```
   SUBROUTINE POST.OFS
* T24 mainline routine
$INSERT I_COMMON
$INSERT I_EQUATE

   ofs_message = 'AB,/I/PROCESS,INPUTT/123456,LISTOFSSRC,ORIGINAL.TEXT::=OFS.SOURCE L'
   ofs_message_id = ''
   CALL OFS.POST.MESSAGE(ofs_message, ofs_message_id, 'TAABS', '')
   CALL JOURNAL.UPDATE(SYSTEM(40))  ;* necessary in mainline routine

   TEXT = ofs_message_id
   CALL REM

   RETURN
END
```

After creating of PGM.FILE record (type M) for this subroutine we can run it:

T24

```
Model Bank R14          POST.OFS


------------------------------------------------------------------------------
------------------------------------------------------------------------------
18 FEB 2015 07:58:41  USER (22 APR) VLADIMIR.K        [28301,ITXN COMPLETE
ACTION
CONTINUE (Y)                                              172162830128725.00
```

See the record in the incoming queue:

jsh

```
CT F.OFS.MESSAGE.QUEUE 172162830128725.00-TAABS
```

Output

```
    172162830128725.00-TAABS
001 AB,/I/PROCESS,INPUTT/123456,LISTOFSSRC,ORIGINAL.TEXT::=OFS.SOURCE L
```

So, actually, except for assigning an @ID all that was done - just a copy of the OFS message was written to the field #1.

To proceed the message set the TSA.SERVICE record BNK/OFS.MESSAGE.SERVICE to START and start tSM. After some time we can see the results:

jsh

```
COUNT F.OFS.MESSAGE.QUEUE

 No Records counted

SELECT F.OFS.RESPONSE.QUEUE LIKE 172162830128725.00...
 1 Records selected

>CT F.OFS.RESPONSE.QUEUE
```

Output

```
    172162830128725.00.1
001 1
002 ORIGINAL.TEXT:1:1=OFS.SOURCE L,RECORD.STATUS:1:1=INAU,CURR.NO:1:1=1,INPUTT
    ER:1:1=11_INPUTTER__OFS_TAABS,DATE.TIME:1:1=1502180811,CO.CODE:1:1=GB00100
    01,DEPT.CODE:1:1=1
003 LISTOFSSRC
```

See the created record:

T24 - ABBREVIATION record

```
     ABBREVIATION.CODE. LISTOFSSRC
 ------------------------------------------------------------------------------
   1 ORIGINAL.TEXT..... OFS.SOURCE L
   2 RECORD.STATUS..... INAU                INPUT Unauthorised
   3 CURR.NO........... 1
   4. 1 INPUTTER....... 11_INPUTTER__OFS_TAABS
   5. 1 DATE.TIME...... 18 FEB 15 08:11
   7 CO.CODE........... GB-001-0001         Model Bank R14
   8 DEPT.CODE......... 1                   Implementation
```

After authorisation we can use this abbreviation to list OFS.SOURCE appplication.

## 1.80 Associated VERSIONs

ASSOC.VERSION field in VERSION record can be used to attach another VERSION screen showing it in additional tab.

Let’s find such VERSIONs for FT:

T24

```
Model Bank R14          VERSION, LIST


------------------------------------------------------------------------------
       SELECTION FIELDS - LIVE FILE      [EQ NE LT GT LE GE RG NR LK UL]
------------------------------------------------------------------------------
0.. @ID                    0A. PGM.NAME.VERSION      1.. PRINT.ONLY
2.. RECORDS.PER.PAGE       3.. FIELDS.PER.LINE       4.. LANGUAGE.CODE
5.. HDR.1.001..039         6.. HDR.1.040..078        7.. HDR.1.079..117
8.. HDR.1.118..132         9.. HDR.2.001..039        10. HDR.2.040..078
11. HDR.2.079..117         12. HDR.2.118..132        13. FIELD.NO
14. COLUMN                 15. EXPANSION             16. TEXT.CHAR.MAX
17. TEXT                   18. TXT.040..078          19. TXT.079..117
20. TXT.118..132           21. ENRICHM.CHAR          22. TABLE.COLUMN
23. TABLE.LINE             24. ENRI.COL              25. PROMPT.COL
26. RESERVED05             27. RESERVED04            28. RESERVED03
29. RESERVED02             30. RESERVED01            31. PROMPT.TEXT
------------------------------------------------------------------------------
@ID                LK FUNDS.TRANSFER,...
ASSOC.VERSION      NE ''
------------------------------------------------------------------------------
19 FEB 2015 05:53:42  USER (22 APR) VLADIMIR.K        [8601,INPPAGE  1 >>4>>>>
ACTION _
ENTER SELECTION (eg CUST EQ 123456) , <F5> TO EXECUTE LIST
```

Result:

T24

```
Model Bank R14           VERSION - Default List

   ID                                                    PRINT.ONLY   FUNCT.
------------------------------------------------------------------------------
 1  FUNDS.TRANSFER,AA.ACCR
 2  FUNDS.TRANSFER,AA.ACCR.CHG.PRIN.WS
 3  FUNDS.TRANSFER,AA.ACCS
 4  FUNDS.TRANSFER,AA.ACCS.CR
 5  FUNDS.TRANSFER,AA.ACDF
 6  FUNDS.TRANSFER,AA.ACDI
 7  FUNDS.TRANSFER,AA.ACDI.DISBURSE.WS
 8  FUNDS.TRANSFER,AA.ACDP
 9  FUNDS.TRANSFER,AA.ACDP.PRECLOSE.PAY.WS
10  FUNDS.TRANSFER,AA.ACDP.REPAY.WS
11  FUNDS.TRANSFER,AA.ACPD
12  FUNDS.TRANSFER,AA.ACPO
13  FUNDS.TRANSFER,AA.ACPO.PAYOFF.WS
14  FUNDS.TRANSFER,AA.ACPP
15  FUNDS.TRANSFER,AA.ACR1
16  FUNDS.TRANSFER,AA.ACR2
------------------------------------------------------------------------------
19 FEB 2015 05:55:55  USER (22 APR) VLADIMIR.K        [8601,INPAGE   1 >>>3>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Scroll down, let’s see the FUNDS.TRANSFER,AUTH one:

T24

```
Model Bank R14          VERSION, SEE

    PGM.NAME.VERSION.. FUNDS.TRANSFER,AUTH
------------------------------------------------------------------------------
 55. 1. 2 VAL.ASSOC... COMMISSION.FOR
 55. 2. 1 VAL.ASSOC... CHARGE.TYPE
 55. 2. 2 VAL.ASSOC... CHARGE.FOR
 55. 3. 1 VAL.ASSOC... TAX.TYPE
 55. 3. 2 VAL.ASSOC... TAX.AMT
 55. 4. 1 VAL.ASSOC... RELATED.MSG
 55. 4. 2 VAL.ASSOC... TIME.IND
 55. 5. 1 VAL.ASSOC... SEND.TO.PARTY
 55. 5. 2 VAL.ASSOC... MESSAGE.TYPE
 55. 6. 1 VAL.ASSOC... MSG.TYPE
 55. 6. 2 VAL.ASSOC... MSG.DATE
 57 LOCAL.REF.FIELD... LOCAL.REF
 65 REPORT.LOCKS...... YES
 67. 1 GB D Authorise/Delete Funds Transfer
 68. 1 ASSOC.VERSION.. FUNDS.TRANSFER,AUDIT Audit                      <=====
 87. 1 ATTRIBUTES..... NO.HEADER.TAB
------------------------------------------------------------------------------
19 FEB 2015 05:58:07  USER (22 APR) VLADIMIR.K        [8601,INPAGE 25  >>26>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Launch this screen in browser...

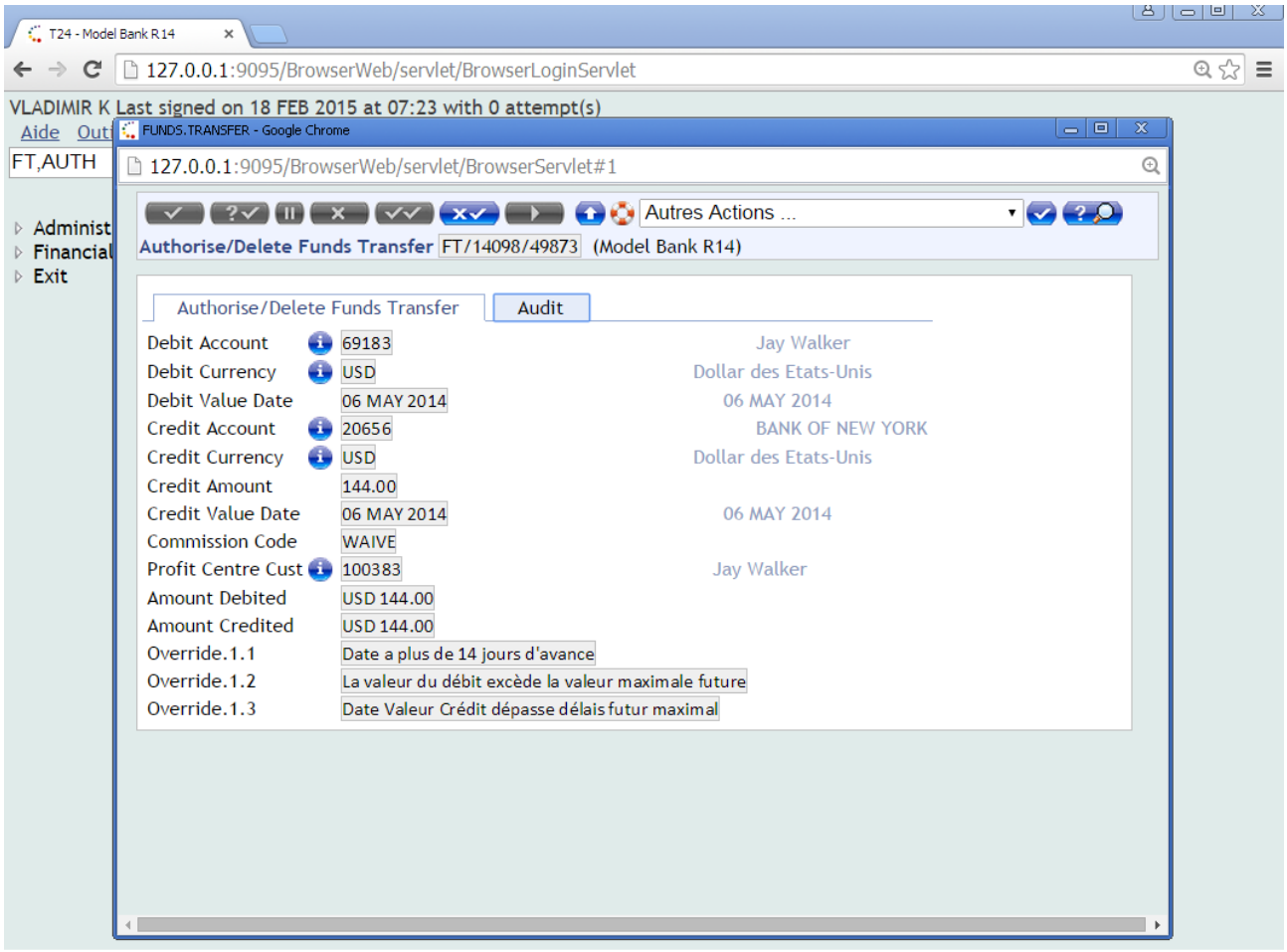


**Figure 47: Multi-tab VERSION, tab 1.**

Click on the second tab...

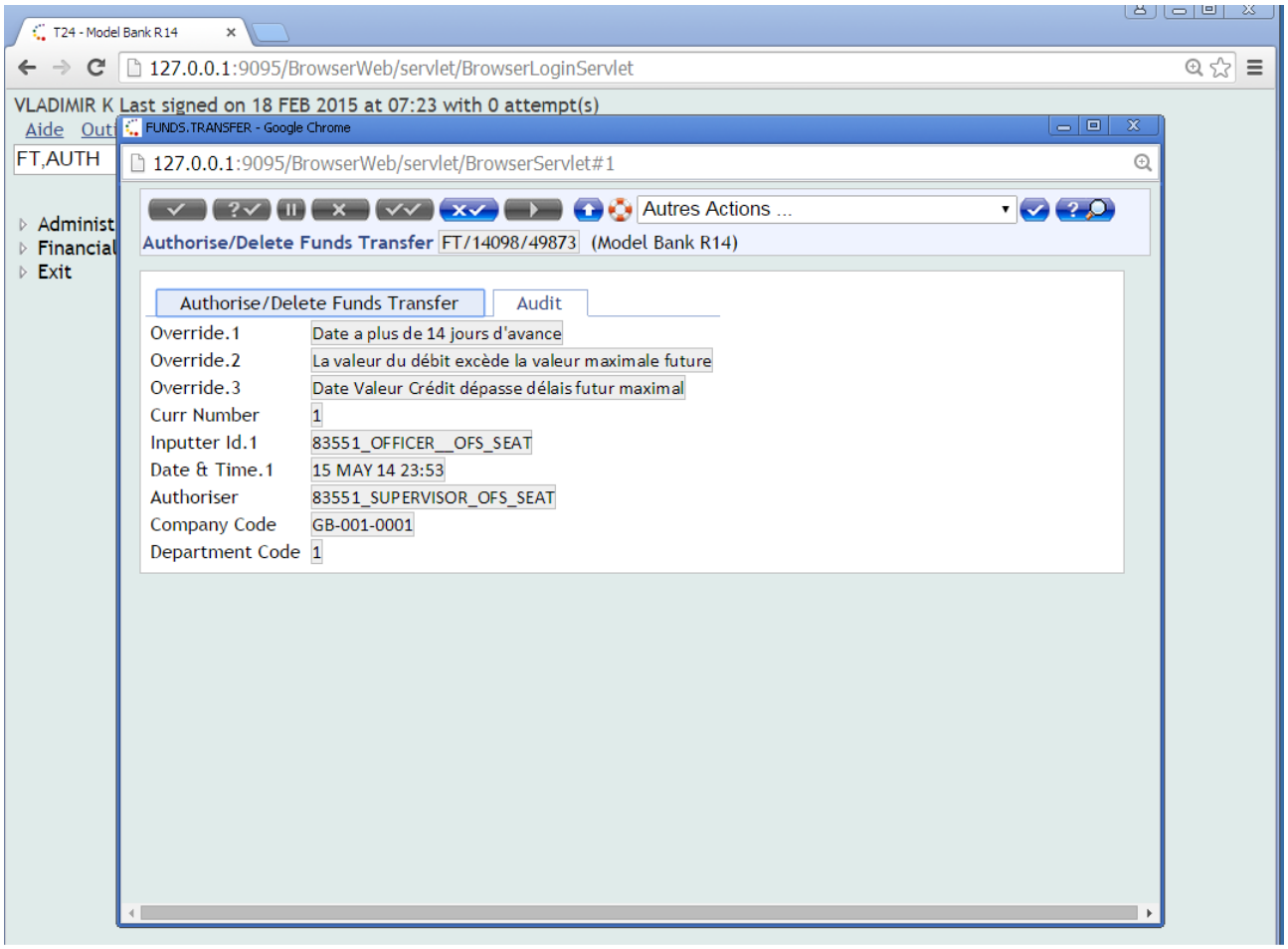


**Figure 48: Multi-tab VERSION, tab 2.**

Associated VERSION is a regular one; it can be launched by itself:

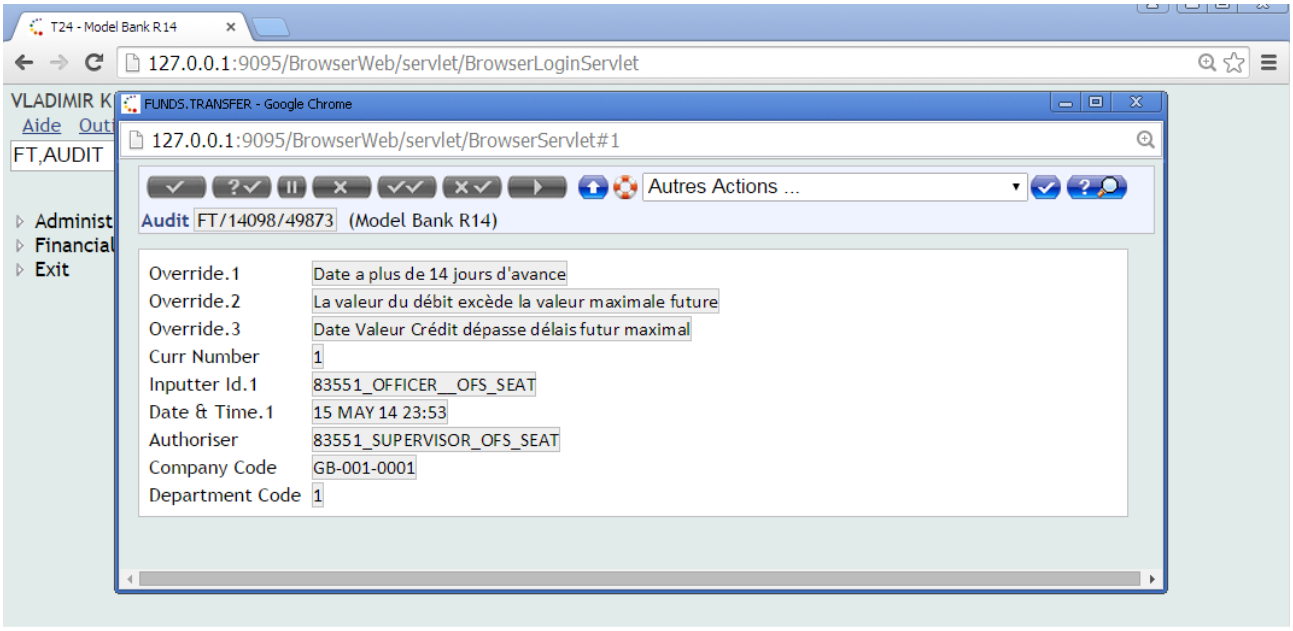


**Figure 49: ASSOC.VERSION by itself.**

Let’s take FT,AUDIT VERSION and attach it to our train VERSION - FT,TRAIN - too and then use it (after commenting debugger in SUBR.ANC otherwise we’ll get session timeout in Browser; at the same routine we also need to correct the call to IN2A: CALL IN2A(’54.3’, ’A’)):

T24

```
Model Bank R14          VERSION, INPUT

    PGM.NAME.VERSION.. FUNDS.TRANSFER,TRAIN
-------------------------------------------------------------------------------
 62 D.SLIP.TRIGGER....
 63. 1 INPUT.ROUTINE.. SUBR.INP
 64. 1 AUTH.ROUTINE ..
 65 REPORT.LOCKS...... YES
 66 GTS.CONTROL.......
 67. 1 GB D
 68. 1 ASSOC.VERSION.. FUNDS.TRANSFER,AUDIT Audit
 69 NEXT.VE _
 70 INITIAL.CURSOR.POS
 71. 1 RESERVED.4.....
 72. 1 CAPTION........
 73 EXC.INC.RTN.......
 74. 1 ID.RTN.........
 75. 1 CHECK.REC.RTN..
 76. 1 AFTER.UNAU.RTN.
 77. 1 BEFORE.AUTH.RTN
-------------------------------------------------------------------------------
19 FEB 2015 06:22:39  USER (22 APR) VLADIMIR.K        [31721,IPAGE 13  >>16>>>
ACTION
```

Now try it:

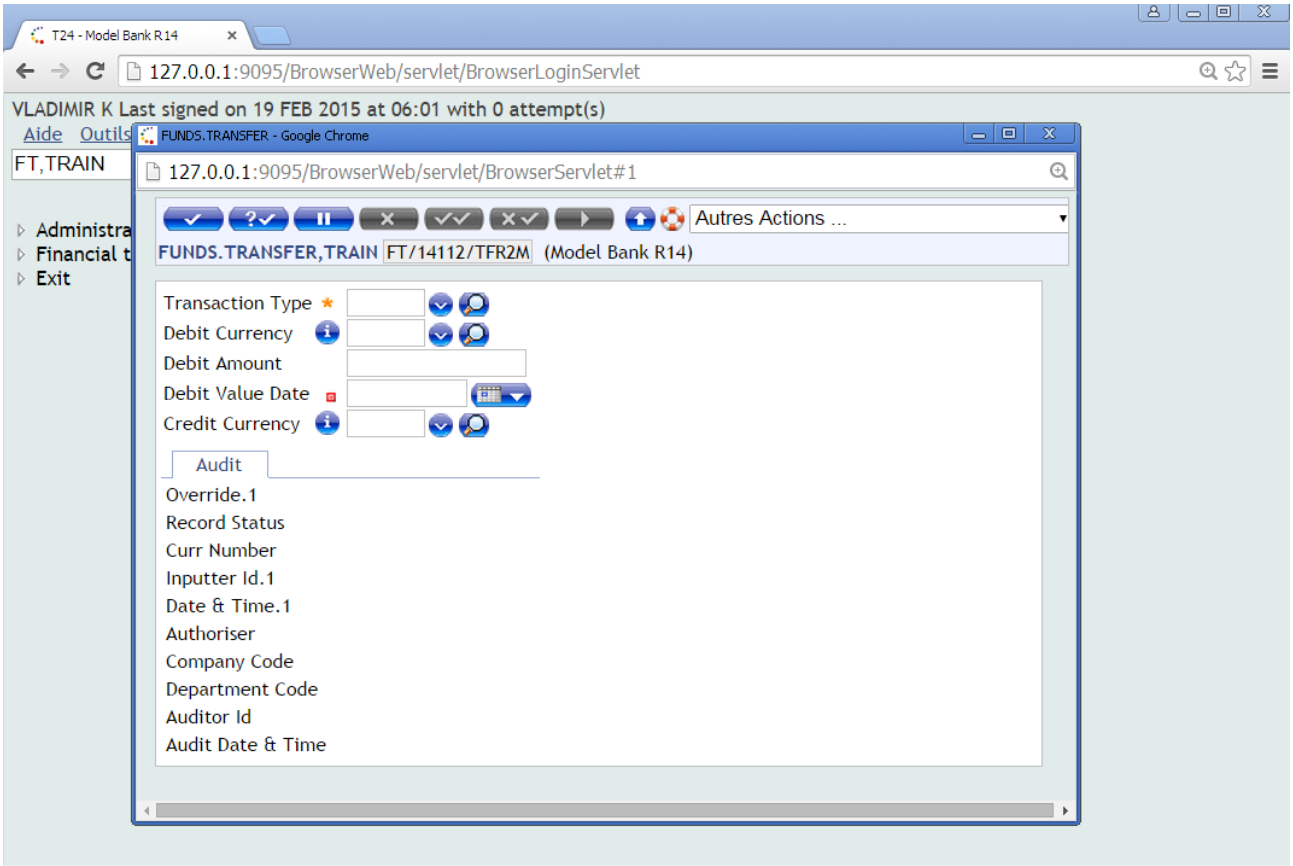


**Figure 50:  Multi-tab VERSION FT,TRAIN.**

In Classic interface additional tabs are not shown:

T24

```
Model Bank R14          FUNDS.TRANSFER,TRAIN INPUT           REF FT14112KSCM3


--------------------------------------------------------------------------------
  1 TRANSACTION.TYPE.. _
  7 DEBIT.VALUE.DATE..
  6 DEBIT.AMOUNT......
  5 DEBIT.CURRENCY....
 13 CREDIT.CURRENCY...




...


--------------------------------------------------------------------------------
19 FEB 2015 06:34:00  USER (22 APR) VLADIMIR.K        [20785,IPAGE 1
ACTION
```

All hook routines (like AUT.NEW.CONTENT, VALIDATION, INPUT etc) should be attached to “main” VERSION, even if auto-populated or validated field isn’t shown on it.

## 1.81 EB.ALTERNATE.KEY

EB.ALTERNATE.KEY application is used to create additional “index” for T24 table:

T24

```
Model Bank R14          EB.ALTERNATE.KEY INPUT

    APPLICATION....... BENEFICIARY         BENEFICIARY
-------------------------------------------------------------------------------
  1 ALT.KEY.MAX.LENGTH 20
  2. 1 ALT.KEY.FIELD.. BEN.ALT.KEY
  3. 1 CONCAT.TYPE.... SYSTEM
  4. 1 ACCESS.METHOD.. READ
  5. 1 UNIQUE......... NONE
  6. 1 ENQUIRY........
  7. 1 BUILD.ROUTINE..
  8. 1 VALIDATION.RTN.
  9 ENRI.FIELD........
 10 RESERVED.14.......
 11 RESERVED.13.......
 12 RESERVED.12.......
 13 RESERVED.11.......
 14 RESERVED.10.......
 15 RESERVED.9........
 16 RESERVED.8........
-------------------------------------------------------------------------------
19 FEB 2015 07:15:02  USER (22 APR) VLADIMIR.K        [13326,IPAGE 1   >>>2>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

In T24 we can use either @ID or the value of BEN.ALT.KEY field to specify BENEFICIARY record; the following screen appears after either of T24 commands:  BENEFICIARY S BEN1409304462 or BENEFICIARY S TEM02:

T24

```
Model Bank R14          BENEFICIARY SEE

    BENEFICIARY.ID.... BEN1409304462
-------------------------------------------------------------------------------
  1. 1 GB NICKNAME.... TOM
  4 CATEGORY.......... Employee
  9 BEN.ACCT.NO....... 66486
 21 TRANSACTION.TYPE.. AC                  Account Transfer
 45. 1 BEN.ALT.KEY.... TEM02
 48 CURR.NO.......... 1
 49. 1 INPUTTER....... 12424_OFFICER__OFS_SEAT
 50. 1 DATE.TIME...... 15 MAY 14 22:02
 51 AUTHORISER........ 12424_OFFICER_OFS_SEAT
 52 CO.CODE........... GB-001-0001         Model Bank R14
 53 DEPT.CODE......... 1                   Implementation
-------------------------------------------------------------------------------
```

So it acts like MNEMONIC here.  To quickly find the necessary record by its BEN.ALT.KEY in T24 subroutine we can use the table where the “index” is stored:

jsh

```
CT F.BENEFICIARY.BEN.ALT.KEY
```

Output

```
    TEM03
001 GB0010001*BEN1409312994

    TEM06
001 GB0010001*BEN1409347996

    TEM01
001 GB0010001*BEN1409314000

    TEM04
001 GB0010001*BEN1409330724

    TEM02
001 GB0010001*BEN1409304462

    TEM05
001 GB0010001*BEN1409394012
```

*(Quotes in “index” mean that it’s not a real jBASE or RDBMS index file but an additional data file for keeping indexed data.  In T24 such files are referred to as “concat” files.)*

Let’s try to create our own alternate key:

T24

```
 Model Bank R14          EB.ALTERNATE.KEY, INPUT

     APPLICATION....... USER                USER PROFILE
 ------------------------------------------------------------------------------
   1 ALT.KEY.MAX.LENGTH
   2. 1 ALT.KEY.FIELD.. INIT.APPLICATION
   3. 1 CONCAT.TYPE.... SYSTEM
   4. 1 ACCESS.METHOD.. READ
   5. 1 UNIQUE......... NONE
   6. 1 ENQUIRY........
   7. 1 BUILD.ROUTINE..
   8. 1 VALIDATION.RTN.
   9 ENRI.FIELD........
  10 RESERVED.14.......
  11 RESERVED.13.......
  12 RESERVED.12.......
  13 RESERVED.11.......
  14 RESERVED.10.......
  15 RESERVED.9........
  16 RESERVED.8........
 ------------------------------------------------------------------------------
 19 FEB 2015 07:32:25  USER (22 APR) VLADIMIR.K        [4135,INPAGE 1   >>>2>>>
 ACTION
```

After commit the concat file is created:

T24

```
 ------------------------------------------------------------------------------
 19 FEB 2015 07:28:40  USER (22 APR) VLADIMIR.K        [4135,INTXN COMPLETE
[ 417 ] File ..\bnk.dict\F_USER_INIT_APPLICATION]D created , type = J4
[ 417 ] File ..\bnk.data\eb\F_USER_INIT_APPLICATION created , type = JR
 AWAITING ID
```

However the file isn’t populated automatically:

jsh

```
COUNT F.USER.INIT.APPLICATION

 No Records counted
```

As soon as this field changes, record in concat file is created or amended.  User VLADIMIR.K1 had INIT.APPLICATION set to ?801.  After clearing this field and setting it back we can see the concat file record:

jsh

```
COUNT F.USER.INIT.APPLICATION

 1 Records counted

CT F.USER.INIT.APPLICATION

    ?801
001 GB0010001*VLADIMIR.K1
```

As this alternate key is not unique (UNIQUE=NONE in EB.ALTERNATE.KEY record), we can assign more users to the same menu (what was intended by this setting).  Let’s do it and check the concat file record again:

jsh

```
CT F.USER.INIT.APPLICATION

    ?801
001 GB0010001*VLADIMIR.K1
002 GB0010001*BUILDUSER0858
```

## 1.82 OFS.GLOBUS.MANAGER

The call to this core T24 routine (which is not recommended but still used by Temenos) allows to process OFS message with immediate effect.  Earlier we used OFS.POST.MESSAGE which just puts the message to the queue for processing.  Here we are able to see the result right away.

We'll use TAABSLOG record in OFS.SOURCE that we copied from TAABS record earlier. OFS message will assign the menu 801 to yet another user (thus illustrating that EB.ALTERNATE.KEY functionality works under OFS as well).

jsh

```
   SUBROUTINE TRAIN.OGM
* Mainline T24 subroutine
$INSERT I_COMMON
$INSERT I_EQUATE

   ofs_msg = 'USER,/I/PROCESS//0,,BUILDUSER1,INIT.APPLICATION::=?801'
   CALL OFS.GLOBUS.MANAGER('TAABSLOG', ofs_msg)

* To show the full output we have to clear the part of screen:
   FOR i = 1 TO 10  ;  CRT ''  ;  NEXT i

* Now show it and wait for a key, otherwise T24 might try to redraw itself
   CRT ofs_msg  ;  INPUT any_key
   RETURN
END
```

*(No user credentials are required in OFS message since we're going to run it already being logged in to T24.)*

OFS.SOURCE type for TAABSLOG has to be set to "GLOBUS"; field IN.QUEUE.DIR is to be populated as well for this type:

T24

```
 Model Bank R14          OFS SOURCE, INPUT

     SOURCE.NAME....... TAABSLOG
 ------------------------------------------------------------------------------
   1 DESCRIPTION....... FOR TAG
   2 SOURCE.TYPE....... GLOBUS
   3. 1 LOGIN.ID....... ANY
   ...
  18 IN.QUEUE.DIR...... TAABSLOGIN
  ...
```

Upon commit the UD file TAABSLOGIN is created (thing that isn't really necessary for us but derived from early Globus days):

T24

```
 19 FEB 2015 08:20:23  USER (22 APR) VLADIMIR.K        [21615,I      VALIDATED [ 417 ]
 File TAABSLOGIN]D created , type = UD
[ 417 ] File TAABSLOGIN created , type = UD                    TXN COMPLETE
1 record(s) deleted.
 AWAITING ID
```

Run TRAIN.OGM from “AWAITING APPLICATION” prompt; note that the question mark in the field INIT.APPLICATION was replaced by a comma:

Output of TRAIN.OGM

```
BUILDUSER1/TLOG150502455430094.00/1,USER.NAME:1:1=BUILDUSER1,SIGN.ON.NAME:1:1=BUILD1,
CLASSIFICATION:1:1=INTERNAL,LANGUAGE:1:1=1,COMPANY.CODE:1:1=GB0010001,DEPARTMENT.CODE
:1:1=1,PASSWORD.VALIDITY:1:1=20141001M0601,START.DATE.PROFILE:1:1=19850101,END.DATE.P
ROFILE:1:1=20151230,START.TIME:1:1=00,END.TIME:1:1=2400,TIME.OUT.MINUTES:1:1=999,ATTE
MPTS:1:1=9,INIT.APPLICATION:1:1=,801,COMPANY.RESTR:1:1=GB0010001,APPLICATION:1:1=ALL.
                                    ^^^^^
PG,FUNCTION:1:1=I L C S P A D E H R V,SIGN.ON.OFF.LOG:1:1=NO,SECURITY.MGMT.L:1:1=NO,A
PPLICATION.LOG:1:1=NO,FUNCTION.ID.LOG:1:1=NO,INPUT.DAY.MONTH:1:1=DDMM,DATE.LAST.SIGN.
ON:1:1=20140611,TIME.LAST.SIGN.ON:1:1=154528,PASSWORD:1:1=F21o9CZEPRXDc4OeYUmOTALOlKW
K7Z4Xm3+xjQ6Nnno=,PASSW.CHANGE.DATE:1:1=20140415,CLEAR.SCREEN:1:1=Y,DEALER.DESK:1:1=0
0,ATTRIBUTES:1:1=SUPER.USER,DATE.FORMAT:1:1=1,SALT:1:1=QDy8TW6wSG,SALT:2:1=g0B7DnDfWy
,SALT:3:1=aJtQnOdIf,CURR.NO:1:1=2,INPUTTER:1:1=24554_VLADIMIR.K__OFS_TAABSLOG,DATE.TI
ME:1:1=1502190821,AUTHORISER:1:1=24554_VLADIMIR.K_OFS_TAABSLOG,CO.CODE:1:1=GB0010001,
DEPT.CODE:1:1=1
```

The comma appeared because it’s standard OFS feature developed for inputting VERSION @IDs etc.  Here we don’t need it so a VERSION hook routine can help here:

jBC

```
SUBROUTINE COMMA.ANC

$INSERT I_COMMON
$INSERT I_EQUATE

   init_app = R.NEW(AF)
   IF init_app[1,1] EQ ',' THEN R.NEW(AF) = '?' : init_app[2,-1]

   RETURN
END
```

After that a VERSION USER,TRAIN is to be created; to be able to use our hook routine there we firstly need to “register” that routine in PGM.FILE:

T24

```
 Model Bank R14          PROGRAM FILE, SEE

     PROGRAM            COMMA.ANC
 ------------------------------------------------------------------------------
   1 TYPE.............. S
   5 PRODUCT........... EB
   8. 1 APPL.FOR.SUBR.. USER               USER PROFILE
  26 CURR.NO........... 1
  27. 1 INPUTTER....... 17496_VLADIMIR.K
  28. 1 DATE.TIME...... 23 FEB 15 06:04
  29 AUTHORISER........ 17496_VLADIMIR.K
  30 CO.CODE........... GB-001-0001        Model Bank R14
  31 DEPT.CODE......... 1                  Implementation
```

VERSION doesn’t need any fields to be described; just a hook attached here:

T24

```
Model Bank R14          VERSION, SEE

    PGM.NAME.VERSION.. USER,TRAIN
--------------------------------------------------------------------------------
  2 RECORDS.PER.PAGE.. 1
  3 FIELDS.PER.LINE... 1
 46 NO.OF.AUTH........ 0
 50. 1 AUTOM.FIELD.NO. INIT.APPLICATION    INIT.APPLICATION
 52. 1 AUT.NEW.CONTENT @COMMA.ANC
```

Also, the mainline routine is to be amended to mention this VERSION in OFS message; “//0” after “PROCESS” is no more necessary since the VERSION has NO.OF.AUTH=0.

The amended line in the source looks as:

jsh

```
...
ofs_msg = 'USER,TRAIN/I/PROCESS,,BUILDUSER1,INIT.APPLICATION::=?801'
...
```

Run it and see that in the output (as well as in the USER record) INIT.APPLICATION field is populated with “?801”:

Output of TRAIN.OGM

```
BUILDUSER1/TLOG150542081422926.00/1,USER.NAME:1:1=BUILDUSER1,SIGN.ON.NAME:1:1=BUILD1,
CLASSIFICATION:1:1=INTERNAL,LANGUAGE:1:1=1,COMPANY.CODE:1:1=GB0010001,DEPARTMENT.CODE
:1:1=1,PASSWORD.VALIDITY:1:1=20141001M0601,START.DATE.PROFILE:1:1=19850101,END.DATE.P
ROFILE:1:1=20151230,START.TIME:1:1=00,END.TIME:1:1=2400,TIME.OUT.MINUTES:1:1=999,ATTE
MPTS:1:1=9,INIT.APPLICATION:1:1=?801, ..............
```

TLOG150542081422926.00 in the reply is the @ID of OFS.REQUEST.DETAIL record:

jsh

```
CT F.OFS.REQUEST.DETAIL 'TLOG150542081422926.00'
```

Output

```
TLOG150542081422926.00
001 USER
002 TRAIN
003 I
004 BUILDUSER1
005
006 GB0010001
007 23 FEB 2015:671 06:22:06
008
009 23 FEB 2015:140 06:22:07
010 PROCESSED
```

```
011 USER,TRAIN/I/PROCESS,/******/,BUILDUSER1,INIT.APPLICATION:=?801
012 BUILDUSER1/TLOG150542081422926.00/1,USER.NAME:1:1=BUILDUSER1,SIGN.ON.NAME:
    1:1=BUILD1,CLASSIFICATION:1:1=INTERNAL,LANGUAGE:1:1=1,COMPANY.CODE:1:1=GB0
    010001,DEPARTMENT.CODE:1:1=1,PASSWORD.VALIDITY:1:1=20141001M0601,START.DAT
    E.PROFILE:1:1=19850101,END.DATE.PROFILE:1:1=20151230,START.TIME:1:1=00,END
    .TIME:1:1=2400,TIME.OUT.MINUTES:1:1=999,ATTEMPTS:1:1=9,INIT.APPLICATION:1:
    1=?801,COMPANY.RESTR:1:1=GB0010001,APPLICATION:1:1=ALL.PG,FUNCTION:1:1=I L
...
    TTER:1:1=20814_VLADIMIR.K__OFS_TAABSLOG,DATE.TIME:1:1=1502230622,AUTHORISE
    R:1:1=20814_VLADIMIR.K_OFS_TAABSLOG,CO.CODE:1:1=GB0010001,DEPT.CODE:1:1=1
013
014
015
016 20814
017
018
019
020
021
022
023
024
025
026 06:22:06:671 23 FEB 2015
027 06:22:07:140 23 FEB 2015
```

Finally, let’s check the concat file:

jsh

```
CT F.USER.INIT.APPLICATION
```

Output

```
    ?801
001 GB0010001*VLADIMIR.K1
002 GB0010001*BUILDUSER0858
003 GB0010001*BUILDUSER1
```

## 1.83 Hook routines for OFS

Besides VERSION hooks that can be used in OFS processing, additional hook routines can be attached to OFS.SOURCE record. The most useful of them are:

• IN.MSG.RTN gets the whole OFS message as a parameter which can be amended and passed for further processing.

• OUT.MSG.RTN gets the whole OFS reply (that we saw on the screen in the previous example); it also can be amended so the OFS-initiating side can get more clear picture of the result.

For example, if we run again the mainline routine TRAIN.OGM we'll get the error:

Output of TRAIN.OGM

```
BUILDUSER1/TLOG150542694424278.00/-1/NO,LIVE RECORD NOT CHANGED
```

The following example will log all such attempts (for example, if we don't have MAINT.MSG.DETS set to Y in OFS.SOURCE we won't have any records created in the file F.OFS.REQUEST.DETAIL but we still might want some track of events):

jsh

```
SUBROUTINE TRAIN.OFS.OUT(ofs_reply)

$INSERT I_COMMON
$INSERT I_EQUATE

   repl_array = ofs_reply           ;* don't touch the reply
   CHANGE '/' TO FM IN repl_array   ;* dynamic array is easier to address

   IF repl_array<3> EQ -1 AND repl_array<4> EQ 'NO,LIVE RECORD NOT CHANGED' THEN
      CALL PRO('ATTEMPT TO OVERWRITE RECORD WITH SAME CONTENT')
   END

   RETURN
END
```

Running OFS message via OFS.GLOBUS.MANAGER doesn't trigger this routine (it might be a core error or an intentional thing).  Test it with tSS (so attach it to the record of OFS.SOURCE that has type TELNET).

Attach the hook (PGM.FILE type S record is necessary for the routine):

T24

```
 Model Bank R14          OFS SOURCE, INPUT

     SOURCE.NAME....... TAABS
 ------------------------------------------------------------------------------
   1 DESCRIPTION....... FOR TAG
   2 SOURCE.TYPE....... TELNET
   3. 1 LOGIN.ID....... ANY
   4. 1 EB.PHANT.ID....
   5 MAX.CONNECTIONS...
   6 RESTRICT.LINK.....
   7 INITIAL.ROUTINE...
   8 CLOSE.ROUTINE.....
   9 IN.MSG.RTN........
  10 OUT.MSG.RTN....... TRAIN.OFS.OUT                          <===== here
  11 MSG.PRE.RTN.......
  12 MSG.POST.RTN......
  13 LOG.FILE.DIR......
  14 LOG.DETAIL.LEVEL.. NONE
  15 OFFLINE.QUEUE.....
  16 MAINT.MSG.DETS....
 ------------------------------------------------------------------------------
```

We need user credentials:

tSS TAABS

```
USER,TRAIN/I/PROCESS,INPUTT/123456,BUILDUSER1,INIT.APPLICATION::=?801
```

Output

```
BUILDUSER1//-1/NO,LIVE RECORD NOT CHANGED
```

CALL PRO writes a record to F.PROTOCOL file; search it:

jsh

```
SEARCH F.PROTOCOL
String :ATTEMPT TO OVERWRITE
String :
Record Keys : *

 2 Records selected
>CT F.PROTOCOL
```

Output

```
    201502231815326838.00
001 20140422
002 1
003 072718
004 07:27:18:890
005 18153
006
007 GB0010001
008 INPUTTER
009 USER,TRAIN
010 1 I
011 BUILDUSER1
012 ATTEMPT TO OVERWRITE RECORD WITH SAME CONTENT
013
014 150223072718890

    INPUTTER-18153
001 20140422
002 1
003 072718
004 07:27:18:890
005 18153
006
007 GB0010001
008 INPUTTER
009 USER,TRAIN
010 1 I
011 BUILDUSER1
012 ATTEMPT TO OVERWRITE RECORD WITH SAME CONTENT
013
014 150223072718890
```

Then - to keep things clean - unhook the routine TRAIN.OFS.OUT from OFS.SOURCE record.

## 1.84 Local overrides

A local override can be appended to core ones (if any) or put to record by itself (if no core ones happened). The INPUT routine for FT,TRAIN VERSION raises the error if debit currency is not the same as credit currency. In the following example we'll amend it to give an override instead of an error.

jBC

```
SUBROUTINE SUBR.INP
* T24 input routine

$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_F.FUNDS.TRANSFER

   dr_ccy = R.NEW(FundsTransfer_DebitCurrency)
   cr_ccy = R.NEW(FundsTransfer_CreditCurrency)

   IF dr_ccy NE cr_ccy THEN
      AF = FundsTransfer_DebitCurrency  ;* set where to return to if reply was NO
      TEXT = "Debit and credit currencies are not the same"
      ove_no = DCOUNT(R.NEW(FundsTransfer_Override), VM) + 1
      CALL STORE.OVERRIDE(ove_no) ;* ove_no=0 will remove all previous overrides,
                                  ;*   including core ones!
   END

   RETURN
END
```

Open appropriate FT record with our VERSION in Browser, click “validate”.

Local override can be seen in the field Override.3:

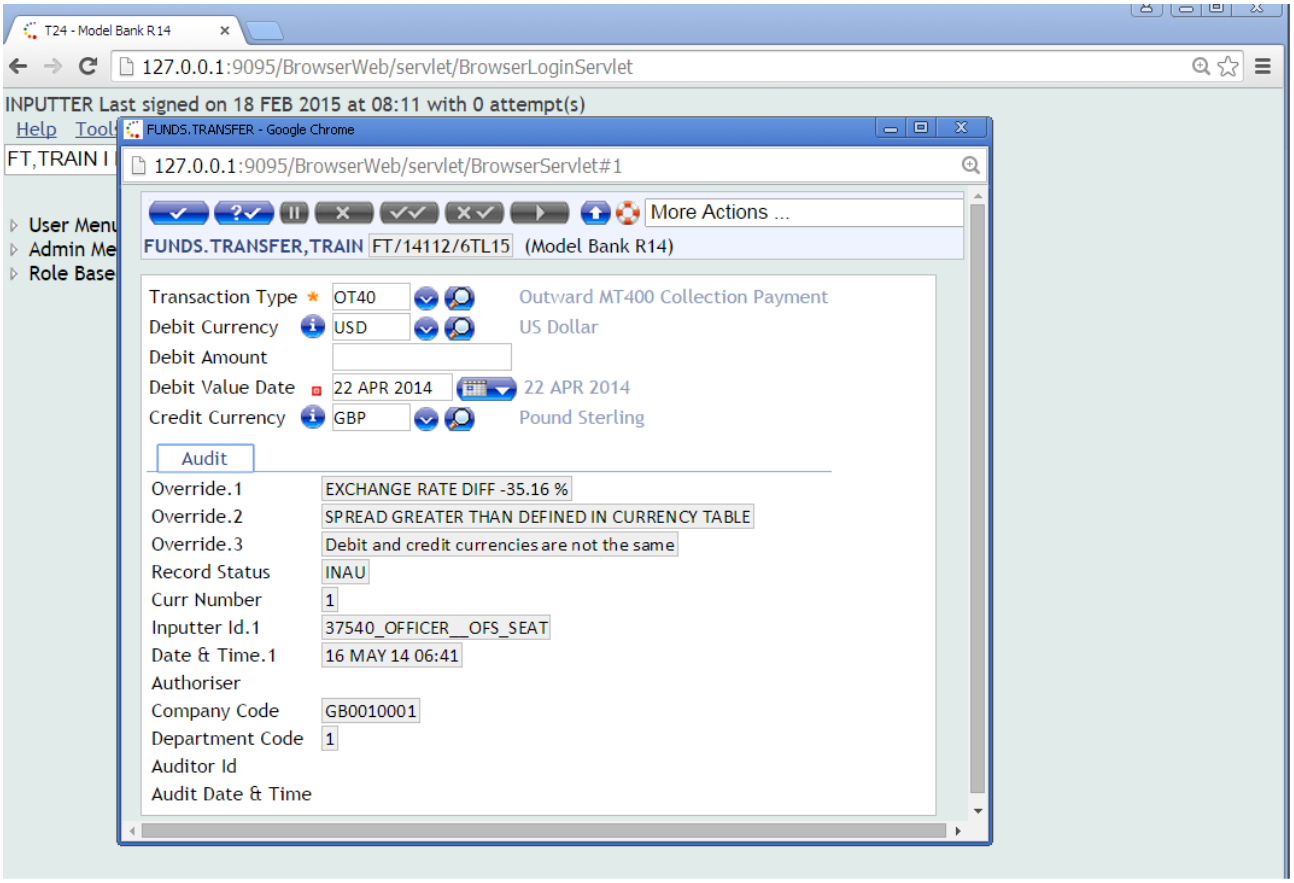


**Figure 51: Local override.**

## 1.85 Sign on routine

A routine that will be triggered upon user login can be specified in USER profile, field SIGN.ON.RTN:

T24

```
Model Bank R14          USER PROFILE, INPUT

    USER.ID.......... VLADIMIR.K1
-------------------------------------------------------------------------------
 ...
 70. 1 SIGN.ON.RTN.... S.O.RTN
-------------------------------------------------------------------------------
```

This routine doesn’t require any parameters; its typical usage - housekeeping tasks (like cleaning of personal local records) or more detailed logging than one that is done when USER field SIGN.ON.OFF.LOG is set to Y.

*(EB.API record is required for this routine.)*

## 1.86 UTF-8 notes

UTF-8 is on if environment variable JBASE_I18N is set to 1 (to correspond with that JBASE_CODEPAGE is to be set to utf8). In T24 routines the global variable RUNNING.IN.UTF8 can be also checked to see if we are under multi-byte characters setup.

When we measure the length of a string, jBC functions LEN() and BYTELEN() return different values if there is at least one multi-byte character in this string. LEN() in this case means “number of characters” rather than bytes.

If there is a non-utf character in a string, an attempt to output it to screen results in fatal error:

UTF-8 error

```
** Error [ UTF8_CONVERSION_FAILED ] A UTF-8 codepage conversion error has occurred.
Source jutil.copycommand.b, Line   418
Press Q to quit
Trap from an error message, error message name = UTF8_CONVERSION_FAILED
Line 418 , Source jutil.copycommand.b
jBASE debugger->
```

If the same command is repeated, jBASE fails but not at the same point but someplace later (caching issue).

If this happens in jbase agent code, the Browser session never gets a reply from the server; JBoss will try to resend the message up to 12 times (each of attempts resulting in error again). User never gets the reply.

The following program writes such a string to &SAVEDLISTS&. OCONV() with string alignment like “10L” (with string longer than 10 characters) puts @TM (text mark - ASCII 251) to its result after every 10 characters (thus mixing utf with non-utf).

Every attempt to output this record to the screen (CT &SAVEDLISTS& UTF-TEST) falls into debugger with conversion error.

jBC

```
utf_line = 'M' : CHAR(399) : 'MM' : CHAR(399) : 'DOV AM' : CHAR(304) : 'D F'   \
    : CHAR(399) : 'RAM' : CHAR(304) : ' ' : CHAR(304) : 'SAYEV SAD' : CHAR(399) \
    : 'DD': CHAR(304) \
    : 'N M' : CHAR(220) : 'RS' : CHAR(399) : 'L'
WRITELIST OCONV(utf_line, '10L') TO 'UTF-TEST'
```

*(Couldn’t reproduce this error under TAFC R14 though if we exchange @TM to, say, CHAR(144) - R14 also fails.... Would like to be able to catch this exception...)*

## 1.87 More about locks

If one user edits particular record, a lock is applied to NAU record (even if it doesn’t yet exist). Other user trying to edit the same record gets the error.

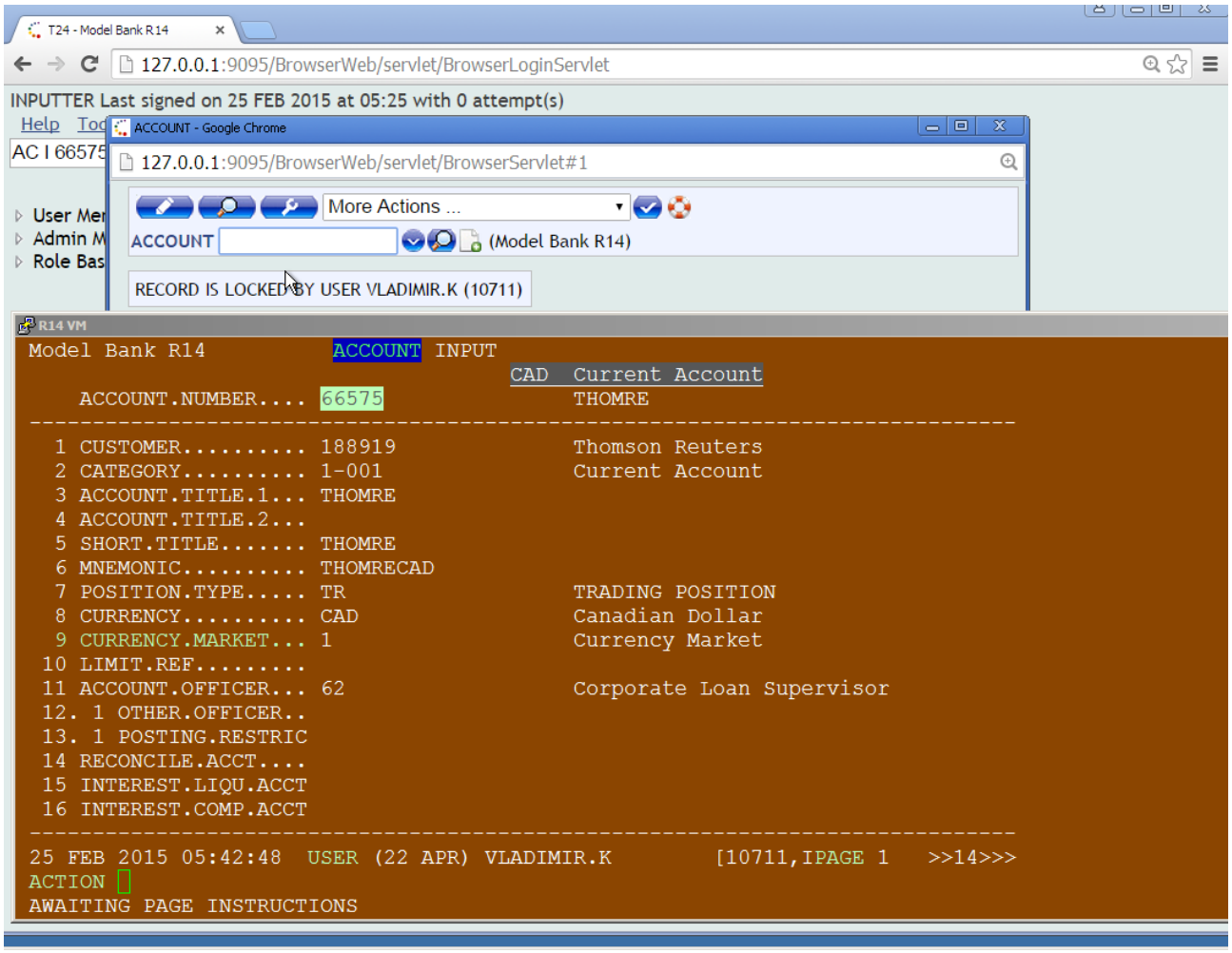


**Figure 52: Record locked by another user.**

Locking information is stored in RECORD.LOCK application:

jsh

```
LIST F.RECORD.LOCK LIKE ...66575
```

Output

```
@ID.............. FBNK.ACCOUNT$NAU.66575
LOCK.KEY........ FBNK.ACCOUNT$NAU.66575
EXPIRE.TIME.....
K.USER.......... VLADIMIR.K
RECORD.STATUS...
CURR.NO......... 1
INPUTTER........ VLADIMIR.K
DATE.TIME....... 1502250552
AUTHORISER...... 1_EB.RECORD.LOCK
CO.CODE......... GB0010001
DEPT.CODE....... 1
AUDITOR.CODE....
AUDIT.DATE.TIME.
FILE.NAME....... F.ACCOUNT$NAU
```

```
RECORD.ID....... 66575
APPLICATION..... ACCOUNT
...
RESERVED.10.....
WINDOW.NAME..... 0
TNO............. 9658
DATE.TIME.SEC... 150225055249
```

If that session terminates abruptly, it could happen that the record is no more edited but the information in F.RECORD.LOCK would be still there so nobody can edit that application record.  To resolve this situation apply V function to RECORD.LOCK (click the button with a triangle directed to the right:

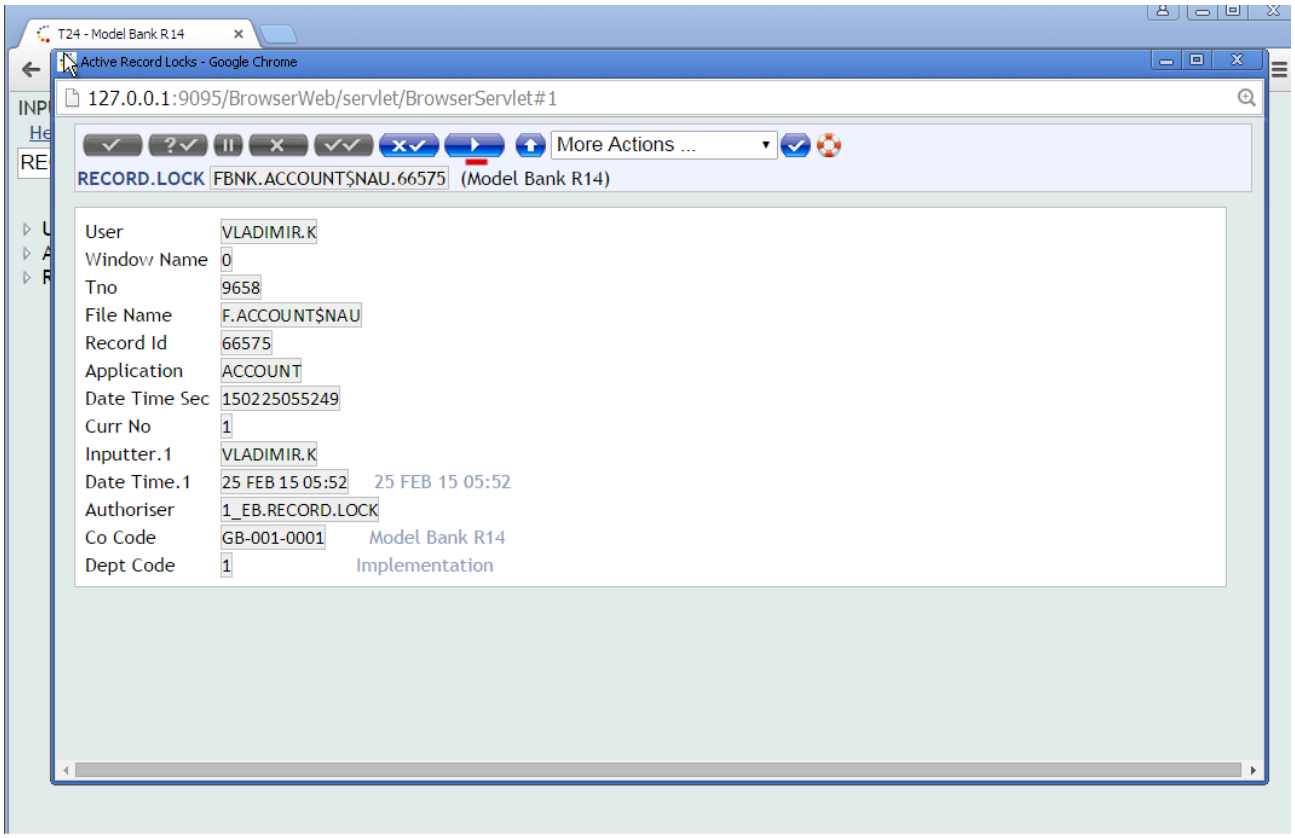


**Figure 53:  Unlock a record.**

After that the record is unlocked....

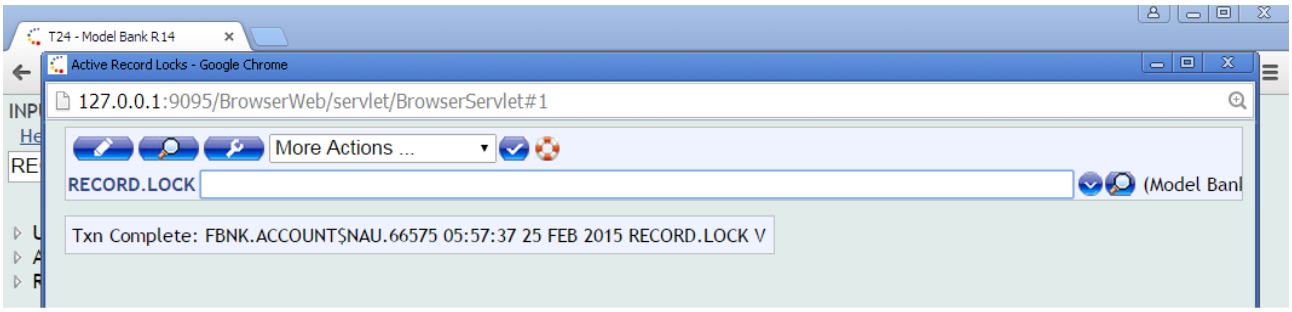


**Figure 54: Record unlocked.**

... and it's no more in RECORD.LOCK application:

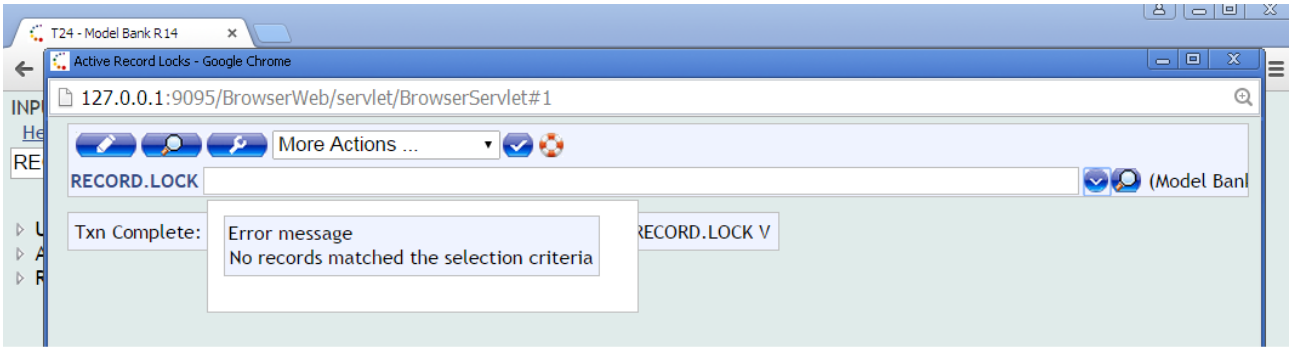


**Figure 55: Lock removed.**

*(Make sure that the session that applied the lock is no more active; always check DATE.TIME field in F.RECORD.LOCK to see if it's a fresh or an old one.)*

## 1.88 Create multi-threaded COB job

Task for this job: roll forward USER field END.DATE.PROFILE.

Step 1. Create 3 subroutines and one insert file:

jBC - insert file

```
* I_ROLL.EDP.COMMON
COMMON /ROLL.EDP.COMMON/ fn$user, f$user, next$year
```

jBC - SELECT routine

```
   SUBROUTINE ROLL.EDP.SELECT
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_ROLL.EDP.COMMON

   curr_year = TODAY[1,4]    ;* will be OK except for the last work day in the year
   year_end = curr_year : '1231'
   sel_cmd = 'SELECT ' : fn$user : ' WITH END.DATE.PROFILE EQ ' : year_end
   EXECUTE sel_cmd RTNLIST user_list CAPTURING output
   CALL BATCH.BUILD.LIST('', user_list)

   RETURN
END
```

jBC - LOAD routine

```
   SUBROUTINE ROLL.EDP.LOAD
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_ROLL.EDP.COMMON

   next$year = TODAY[1,4] + 1  ;* will be OK except for the last work day in the year
   fn$user = 'F.USER'  ;  f$user = ''
   CALL OPF(fn$user, f$user)

   RETURN
END
```

jBC - RECORD routine

```
SUBROUTINE ROLL.EDP(user_id)
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_ROLL.EDP.COMMON

   ofs_message = 'USER,/I/PROCESS//0,INPUTT/123456,' : user_id
   ofs_message := ',END.DATE.PROFILE::=' : next$year : '1231'
   ofs_message_id = ''
   CALL OFS.POST.MESSAGE(ofs_message, ofs_message_id, 'TAABS', '')
   CALL OCOMO('Posted ofs message ' : ofs_message_id)

   RETURN
END
```

Step 2.  Create necessary records in T24 applications.

PGM.FILE:

T24

```
Model Bank R14          PROGRAM FILE, SEE

    PROGRAM            ROLL.EDP
-------------------------------------------------------------------------------
  1 TYPE.............. B
  4. 1 BATCH.JOB...... @BATCH.JOB.CONTROL
  5 PRODUCT........... EB
 26 CURR.NO........... 1
 27. 1 INPUTTER....... 23846_VLADIMIR.K
 28. 1 DATE.TIME...... 26 FEB 15 06:22
 29 AUTHORISER........ 23846_VLADIMIR.K
 30 CO.CODE........... GB-001-0001         Model Bank R14
 31 DEPT.CODE......... 1                   Implementation
```

BATCH:

T24

```
Model Bank R14          BATCH ENTRY, INPUT

        BATCH PROCESS..... BNK/COB.TRAIN
    ---------------------------------------------------------------------------
  1 BATCH.STAGE....... A100
  2 DEFAULT.PRINTER...
  3 PROCESS.STATUS.... 0
  4 BATCH.ENVIRONMENT. F
  5 DEPARTMENT.CODE...
  6. 1 JOB.NAME....... ROLL.EDP
  7. 1. 1 VERIFICATION
  8. 1 FREQUENCY...... A
  9. 1 NEXT.RUN.DATE.. 22 APR 2014
```

Check the results after running COB:

COMO contents

```
..COB.TRAIN_ROLL.EDP_3_26 FEB 2015_06:38:49:687_Posted ofs message 172240536523929.00
..COB.TRAIN_ROLL.EDP_3_26 FEB 2015_06:38:49:703_Posted ofs message 172240536523929.01
..COB.TRAIN_ROLL.EDP_3_26 FEB 2015_06:38:49:703_Posted ofs message 172240536523929.02
..COB.TRAIN_ROLL.EDP_3_26 FEB 2015_06:38:49:718_Posted ofs message 172240536523929.03
..COB.TRAIN_ROLL.EDP_3_26 FEB 2015_06:38:49:718_Posted ofs message 172240536523929.04
```

jsh

```
CT F.OFS.MESSAGE.QUEUE
```

Output

```
    172240536523929.00-TAABS
001 USER,/I/PROCESS//0,INPUTT/123456,BUILDUSER98,END.DATE.PROFILE::=20151231

    172240536523929.03-TAABS
001 USER,/I/PROCESS//0,INPUTT/123456,BUILDUSER34,END.DATE.PROFILE::=20151231

    172240536523929.01-TAABS
001 USER,/I/PROCESS//0,INPUTT/123456,BUILDUSER40,END.DATE.PROFILE::=20151231

    172240536523929.02-TAABS
001 USER,/I/PROCESS//0,INPUTT/123456,BUILDUSER8200,END.DATE.PROFILE::=20151231

    172240536523929.04-TAABS
001 USER,/I/PROCESS//0,INPUTT/123456,SCCO,END.DATE.PROFILE::=20151231
```

Start OFS.MESSAGE.SERVICE to proceed the records.  Check the result for the last record listed above:

jsh

```
SELECT F.OFS.RESPONSE.QUEUE LIKE 172240536523929.04...

 1 Records selected

>CT F.OFS.RESPONSE.QUEUE
```

Output

```
    172240536523929.04.1
001 1
002 USER.NAME:1:1=SC Corporate Actions User,SIGN.ON.NAME:1:1=SCCUSER,CLASSIFIC
    ATION:1:1=INT,LANGUAGE:1:1=1,COMPANY.CODE:1:1=GB0010001,DEPARTMENT.CODE:1:
    1=14,PASSWORD.VALIDITY:1:1=20141101M0601,START.DATE.PROFILE:1:1=20140515,E
    ND.DATE.PROFILE:1:1=20151231,START.TIME:1:1=0,END.TIME:1:1=2400,TIME.OUT.M
    INUTES:1:1=999,ATTEMPTS:1:1=9,COMPANY.RESTR:1:1=GB0010001,APPLICATION:1:1=
    ALL.PG,FUNCTION:1:1=A 2 B C D E F H I L P R S V ,SIGN.ON.OFF.LOG:1:1=NO,SE
    CURITY.MGMT.L:1:1=NO,APPLICATION.LOG:1:1=NO,FUNCTION.ID.LOG:1:1=NO,INPUT.D
    AY.MONTH:1:1=DDMM,DATE.LAST.SIGN.ON:1:1=20140520,TIME.LAST.SIGN.ON:1:1=072
    727,PASSWORD:1:1=pHXHpxxb3l1n+YAKYoTildNygz5ooxe1alEGxtopHd8=,PASSW.CHANGE
    .DATE:1:1=20140520,CLEAR.SCREEN:1:1=Y,DEALER.DESK:1:1=00,ATTRIBUTES:1:1=SU
    PER.USER,DATE.FORMAT:1:1=1,SALT:1:1=yG6MxrCuz,CURR.NO:1:1=2,INPUTTER:1:1=2
    _INPUTTER__OFS_TAABS,DATE.TIME:1:1=1502260647,AUTHORISER:1:1=2_INPUTTER_OF
    S_TAABS,CO.CODE:1:1=GB0010001,DEPT.CODE:1:1=1
003 SCCO
```

We can convert this or other multi-threaded COB job to online service in 2 easy steps.

Step 1.  Clear the value of BATCH.STAGE field in corresponding BATCH record (and - to be on the safe side - make it “daily”):

T24

```
Model Bank R14          BATCH ENTRY, INPUT

    BATCH PROCESS..... BNK/COB.TRAIN
------------------------------------------------------------------------------
  1 BATCH.STAGE.......                                                  <=====
  2 DEFAULT.PRINTER...
  3 PROCESS.STATUS.... 0
  4 BATCH.ENVIRONMENT. F
  5 DEPARTMENT.CODE...
  6. 1 JOB.NAME....... ROLL.EDP
  7. 1. 1 VERIFICATION
  8. 1 FREQUENCY...... D                                               <=====
  9. 1 NEXT.RUN.DATE..                                                 <=====
 10. 1 PRINTER.NAME...
 11. 1. 1 DATA........
 12. 1 JOB.STATUS..... 0
 13. 1 LAST.RUN.DATE.. 22 APR 2014
 14. 1 JOB.MESSAGE....
 15. 1 USER...........
 16. 1 SELECT.AHEAD...
------------------------------------------------------------------------------
26 FEB 2015 06:53:52  USER (22 APR) VLADIMIR.K        [31026,IPAGE 1   >>>3>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Step 2.  Create TSA.SERVICE record for our service:

T24

```
Model Bank R14          TSA.SERVICE, INPUT

    SERVICE........... BNK/COB.TRAIN
------------------------------------------------------------------------------
  1. 1 DESCRIPTION.... TRAIN
  2. 1 SERVER.NAME....
  3. 1 WORK.PROFILE... OFS.MESSAGE.SERVICE OFS TSA message service
  4. 1 SERVER.STATUS..
  5 USER.............. INPUTTER            INPUTTER
  6 SERVICE.CONTROL... STOP
  7 REVIEW.TIME.......
  8 TIME.OUT..........
  9. 1 ATTRIBUTE.TYPE.
 10. 1 ATTRIBUTE.VALUE
 11 FREQUENCY.........
 12 RESERVED.8........
 13 RESERVED.7........
 14. 1 LOCAL.REF......
 15. 1 DATE........... 22 APR 2014
 16. 1 STARTED........ 26/02/2015 06:35:57
------------------------------------------------------------------------------
26 FEB 2015 06:54:54  USER (22 APR) VLADIMIR.K        [31026,IPAGE 1   >>>3>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

## 1.89 User interaction in Classic mode

In Browser there’s no any possibility of interacting with user in jBC. In Classic, however, it’s possible should some technical routine requires that. We already used CALL REM in mainline routine to output a message and exit when user inputs Y.

The following examples also refer to T24 mainline routine (PGM.FILE type M).

Ask a question with possibility for the user to reply Y/NO:

jBC - mainline routine

```
total_qty = 42
TEXT = 'Found & messages. Continue?' :FM: total_qty
CALL OVE
IF TEXT NE 'Y' THEN RETURN
```

User screen:

T24

```
 -- LAST SIGN.ON, DATE: 26 FEB 2015     TIME: 06:53        ATTEMPTS: 0 --------
 26 FEB 2015 07:30:15  USER (22 APR) VLADIMIR.K        [9987,IN]
 ACTION _
 OVERRIDE (Y/NO)                                    Found 42 messages. Continue?
```

Ask to input some data, e.g. list name to proceed:

jBC - mainline routine

```
CALL TXTINP('Input list name or ENTER to cancel', 10, 22, '64', 'ANY')
IF COMI EQ '' THEN
   TEXT = 'Cancelled'  ;  CALL REM  ;  RETURN
END
```

User screen:

T24

```
 ------------------------------------------------------------------------------
 26 FEB 2015 07:32:25  USER (22 APR) VLADIMIR.K        [20679,IN]
 ACTION   _
 Input list name or ENTER to cancel
```

After pressing Enter:

T24

```
--------------------------------------------------------------------------------
26 FEB 2015 07:32:25  USER (22 APR) VLADIMIR.K        [20679,IN]
ACTION _
CONTINUE (Y)                                                        Cancelled
```

Output something to screen (e.g.  progress gauge) without waiting for user input:

jBC

```
   SUBROUTINE SUBR.0
* T24 mainline routine
$INSERT I_COMMON
$INSERT I_EQUATE

   qty = 30000
   FOR i = 1 TO qty
      IF MOD(i, 1000) EQ 0 THEN
         the_done = INT(i * 13 / qty)
         CALL DISPLAY.MESSAGE('-> ' :    \
            STR('+', the_done) : STR('.', 13 - the_done), 2)
         MSLEEP(500)       ;* not to be too fast
      END
   NEXT i

   RETURN
END
```

Screen:

T24

```
--------------------------------------------------------------------------------
19 MAR 2015 13:10:24  USER (22 APR) VLADIMIR.K        [27857,I-> ++++++.......
ACTION
```

## 1.90 Frequency fields

There is special type of fields in T24 that are called “frequency” fields (nothing in common with frequency of COB jobs).

See the field PASSWORD.VALIDITY in USER record:

T24

```
Model Bank R14          USER PROFILE INPUT

    USER.ID........... AUTHORISER
------------------------------------------------------------------------------
  1 USER.NAME......... AUTHORISER
  2 SIGN.ON.NAME...... AUTHOR
  3 CLASSIFICATION.... INTERNAL
  4 LANGUAGE.......... 1                   English
  5. 1 COMPANY.CODE... GB0010001           Model Bank R14
  5. 2 COMPANY.CODE... GB0010002           MF LEAD COMPANY
  5. 3 COMPANY.CODE... EU0010001           Model Bank - Europe
  5. 4 COMPANY.CODE... SG0010001           Model Bank - Singapore
  5. 5 COMPANY.CODE... GB0010005           Model Bank Branch 1
  5. 6 COMPANY.CODE... GB0010003           MF Branch 1
  5. 7 COMPANY.CODE... GB0010004           MF Branch 2
  6 DEPARTMENT.CODE... 1                   Implementation
  7 PASSWORD.VALIDITY. 28 MAR 2015 M0601   28 MAR 2015 Every 6 months on day 1
  8 START.DATE.PROFILE 28 MAR 2014
  9 END.DATE.PROFILE.. 28 MAR 2020
 10. 1 START.TIME..... 00:00
------------------------------------------------------------------------------
06 MAR 2015 10:58:23  USER (22 APR) VLADIMIR.K        [16045,IPAGE 1   >>>6>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

In the data record they are stored like:

jsh

```
LIST F.USER 'AUTHORISER' PASSWORD.VALIDITY
```

Output

```
@ID.............    PASSWORD.VALIDITY

AUTHORISER          20150328M0601
```

If we reset the password for this user...

T24

```
Model Bank R14          PASSWORD.RESET, INPUT

    KEY.............. 1
--------------------------------------------------------------------------------
  1. 1 USER.PW.ATTEMPT
  2. 1 USER.ATTEMPT...
  3. 1 NO.OF.ATTEMPTS.
  4. 1 USER.DEACT.PERD
  5. 1 DEACTIV.PERIOD.
  6. 1 USER.RESET..... AUTHORISER
  7. 1 USER.PASSWORD.. 123456
  8. 1 USER
  9 USER.TYPE......... INT
 10 RECORD.STATUS.....
 11 CURR.NO...........
 12. 1 INPUTTER.......
 13. 1 DATE.TIME......
 14 AUTHORISER........
 15 CO.CODE...........
 16 DEPT.CODE.........
--------------------------------------------------------------------------------
06 MAR 2015 11:21:13  USER (22 APR) VLADIMIR.K        [24427,IPAGE 1   >>>2>>>
ACTION
```

...PASSWORD.VALIDITY gets the value of today’s system date:

jsh

```
LIST F.USER 'AUTHORISER' PASSWORD.VALIDITY
```

Output

```
@ID.............    PASSWORD.VALIDITY

AUTHORISER          20150305M0601
```

Then, when this user logs in to T24, the message appears:

T24

```
--------------------------------------------------------------------------------
06 MAR 2015 11:26:12  USER                              [18743,IN]
ACTION _
PASSWORD TERMINATED, ENTER NEW ONE
```

Upon password change the value of PASSWORD.VALIDITY field becomes:

jsh

```
LIST F.USER 'AUTHORISER' PASSWORD.VALIDITY
```

Output

```
@ID.............    PASSWORD.VALIDITY

AUTHORISER          20150901M0601
```

There is T24 API subroutine CFQ that can be called to calculate the next date for given frequency.  The following jBC program takes the date 20150305 from the example above and rolls it forward twice according to the defined frequency M0601:

jBC program

```
$INSERT I_COMMON
$INSERT I_EQUATE

   test_date = '20150305M0601'
   GOSUB ROLL.DATE
   GOSUB ROLL.DATE

   STOP

ROLL.DATE:
   COMI = test_date
   CALL CFQ
   test_date = COMI
   CRT test_date
   RETURN
```

Output

```
20150901M0601
20160301M0601
```

*(Note that we have a PROGRAM here rather than a SUBROUTINE but still need I_COMMON and I_EQUATE to be able to address the global T24 variable COMI.)*

Custom frequency can be defined in T24 application EB.FREQUENCY, see the example of existing record:

T24

```
 Model Bank R14          EB.FREQUENCY, INPUT

     FREQ.ID.......... LHFYR
 ------------------------------------------------------------------------------
   1. 1 GB DESCRIPTION. LAST DAY IN THIS MID YEAR
   2. 1 FREQ.DATE......
   3 SPECIAL.ROUTINE... LAST.DAY.OF.THIS.HFYR LAST.DAY.OF.THIS.HFYR
   4 NO.OF.PAYMENTS....
   5 RESERVED.4........
   6 RESERVED.3........
   7 RESERVED.2........
   8 RESERVED.1........
   9 NO.DAYS.MAINTAIN..
```

Let’s create our own frequency - e.g.  use only February 29...

Step 1.  The subroutine:

jBC

```
   SUBROUTINE LEAP.YR.ONLY
* T24 custom frequency routine
$INSERT I_COMMON
$INSERT I_EQUATE

   the_date = COMI[1,8]  ;  the_year = the_date[1,4]  ;  the_tail = the_date[5,4]
   the_fqy = COMI[9,5]  ;  target_day = '0229'
   IF the_tail GE target_day THEN the_year++   ;* start from the next year

   LOOP
      COMI = the_year : target_day
      CALL IN2D('D', 11)
      IF ETEXT EQ '' THEN BREAK
      the_year++
   REPEAT

   COMI := the_fqy

   RETURN
END
```

Step 2.  Create EB.FREQUENCY record (requires PGM.FILE type S record for routine LEAP.YR.ONLY):

T24

```
Model Bank R14          EB.FREQUENCY, INPUT

    FREQ.ID........... LEAPY
------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. 29 FEB ONLY
  2. 1 FREQ.DATE......
  3 SPECIAL.ROUTINE... LEAP.YR.ONLY
  4 NO.OF.PAYMENTS....
  5 RESERVED.4........
```

Test it.  Trying to set PASSWORD.VALIDITY to LEAPY results in an override:

T24

```
------------------------------------------------------------------------------
06 MAR 2015 12:27:16  USER (22 APR) VLADIMIR.K        [16813,IPAGE 1   >>>6>>>
ACTION _
OVERRIDE (Y/NO)                                        DATE > TODAY + 6 MONTHS
```

After confirmation we see the result:

T24

```
 Model Bank R14          USER PROFILE, INPUT

     USER.ID........... AUTHORISER
 ------------------------------------------------------------------------------
   1 USER.NAME......... AUTHORISER
   2 SIGN.ON.NAME...... AUTHOR
   3 CLASSIFICATION.... INTERNAL
   4 LANGUAGE.......... 1                   English
   5. 1 COMPANY.CODE... GB0010001           Model Bank R14
   5. 2 COMPANY.CODE... GB0010002           MF LEAD COMPANY
   5. 3 COMPANY.CODE... EU0010001           Model Bank - Europe
   5. 4 COMPANY.CODE... SG0010001           Model Bank - Singapore
   5. 5 COMPANY.CODE... GB0010005           Model Bank Branch 1
   5. 6 COMPANY.CODE... GB0010003           MF Branch 1
   5. 7 COMPANY.CODE... GB0010004           MF Branch 2
   6 DEPARTMENT.CODE... 1                   Implementation
   7 PASSWORD.VALIDITY. 29 FEB 2016 LEAPY   29 FEB 2016 29 FEB ONLY        <======
   8 START.DATE.PROFILE 28 MAR 2014
   9 END.DATE.PROFILE.. 28 MAR 2020
  10. 1 START.TIME..... 00:00
 ------------------------------------------------------------------------------
 06 MAR 2015 12:27:16  USER (22 APR) VLADIMIR.K        [16813,IPAGE 1   >>>6>>>
 ACTION
```

For the final test let’s amend the jBC program where we used CFQ API routine - let it work with our new frequency:

jBC

```
$INSERT I_COMMON
$INSERT I_EQUATE

   test_date = '19990305LEAPY'
   GOSUB ROLL.DATE
   test_date = '20120101LEAPY'
   GOSUB ROLL.DATE
   test_date = '20120229LEAPY'
   GOSUB ROLL.DATE
   test_date = '20150305LEAPY'
   GOSUB ROLL.DATE

   STOP

ROLL.DATE:
   COMI = test_date
   CALL CFQ
   CRT COMI
RETURN
```

Output

```
20000229LEAPY
20120229LEAPY
20160229LEAPY
20160229LEAPY
```

## 1.91 Duplicate values control

Many T24 applications have the control of duplicate values in some fields, e.g. for OTHER.OFFICER below:

T24

```
Model Bank R14          CUSTOMER INPUT

    CUSTOMER.CODE..... 129017
--------------------------------------------------------------------------------
 17. 1 REL.RESERV6....
 18. 1 REL.RESERV5....
 19. 1 REL.RESERV4....
 20. 1 REL.RESERV3....
 21. 1 REL.RESERV2....
 22. 1 REL.RESERV1....
 23 SECTOR............ 1001               Individual
 24 ACCOUNT.OFFICER... 29                 Retail Credit Manager
 25. 1 OTHER.OFFICER.. 18                 Retail Loan Administrator 1
 25. 2 OTHER.OFFICER.. 18                                        DUPLICATE
 26 INDUSTRY.......... 1000               Private Person (Name)
 27 TARGET............ 1                  Private Client - Standard
 28 NATIONALITY....... US                 United States of America
 29 CUSTOMER.STATUS... 1                  Private Client - Standard
 30 RESIDENCE......... US                 United States of America
 31 CONTACT.DATE...... 22 APR 2014
--------------------------------------------------------------------------------
24 MAR 2015 10:15:31  USER (22 APR) VLADIMIR.K        [13545,IPAGE 2   >>12>>>
ACTION
```

To provide such a check in local code use the T24 API subroutine DUP.

However it checks values in particular field and if we have a more complex situation like in DL.DEFINE where we can accidentally save the same record 2 times...

T24

```
 Model Bank R14          DL.DEFINE INPUT

     UNIT.NAME......... TMNS000-QWE
 ------------------------------------------------------------------------------
   1. 1. 1 GB DESCRIPTN SAVE #2
   2. 1 GB SHORT.DESC.. SAVE 2
   3 LANGUAGE/COUNTRY..
   4. 1 INDICES........
   5 OPERATION......... S
   6 SELECT.LIST.......
   7 TOP.LEVEL.TYPE....
   8 TOP.LEVEL.ITEM....
   9. 1 FILE.NAME...... VERSION                    <=====
  10. 1 RECO FUNDS.TRANSFER,TRAIN                  <=====
  11. 1. 1 RECORD.DESC.
   9. 2 FILE.NAME...... ENQUIRY
  10. 2 RECO TRAIN.ENQ
  11. 2. 1 RECORD.DESC.
   9. 3 FILE.NAME...... VERSION                    <=====
  10. 3 RECO FUNDS.TRANSFER,TRAIN                  <=====
 ------------------------------------------------------------------------------
 24 MAR 2015 10:28:25  USER (22 APR) VLADIMIR.K        [13545,IPAGE 1   >>>3>>>
 ACTION _
 AWAITING PAGE INSTRUCTIONS
```

...we need an INPUT routine to check the combination of 2 fields.  However, it looks like INPUT routine isn’t triggered for either this particular application or for all “W” templates so instead we can use CHECK.REC.RTN that will apply this check (for function “V” only, otherwise we wouldn’t be able to correct such record).

jBC

```
    SUBROUTINE DL.DEF.DUP
$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_F.DL.DEFINE
   IF V$FUNCTION NE 'V' THEN RETURN
   chk_array = ''
   val_qty = DCOUNT(R.NEW(DL.DEF.FILE.NAME), @VM)

   FOR i = 1 TO val_qty
      chk_val = R.NEW(DL.DEF.FILE.NAME)<1,i> : '///' : R.NEW(DL.DEF.RECORD.NAME)<1,i>
      FIND chk_val IN chk_array SETTING found ELSE found = ''
      IF found THEN
         E = 'DUPLICATE SET #&' :FM: i
         BREAK
      END ELSE
         chk_array<-1> = chk_val
      END
   NEXT i
   RETURN
END
```

Try to use V function:

T24

```
Model Bank R14          DL.DEFINE,TRAIN VERIFY


------------------------------------------------------------------------------









------------------------------------------------------------------------------
24 MAR 2015 11:09:09  USER (22 APR) VLADIMIR.K        [16911,IN]
ACTION QWE                                                  DUPLICATE SET #3
AWAITING ID
```

## 1.92 Copying report files via T24 printing subsystem

Here - the example how it is set for AutoFORMLaserNet in R14.

Step 1. PRINTER.ID:

T24

```
Model Bank R14          PRINTER FILE MAINTENANCE SEE

    PRINTER.NAME...... AUTOFORMLASERNET
------------------------------------------------------------------------------
  1 PRIME.PRINTER.ID.. AUTOFORMLASERNET
  3 PRINTER.TYPE...... FILEPRINT
  4. 1. 1 COMMAND..... DOS /c 'MOVE.BAT &HOLD.CONTROL>@ID&'
 12 CURR.NO........... 1
 13. 1 INPUTTER....... 1094_AUTHORISER__OFS_BROWSERTC
 14. 1 DATE.TIME...... 31 MAY 14 10:53
 15 AUTHORISER........ 680_INPUTTER_OFS_BROWSERTC
 16 CO.CODE........... GB-001-0001         Model Bank R14
 17 DEPT.CODE......... 1                   Implementation
------------------------------------------------------------------------------
```

Let’s see what’s there in MOVE.BAT mentioned in COMMAND field:

jsh

```
CT . MOVE.BAT
```

Output

```
    MOVE.BAT
001 copy ..\bnk.data\eb\"&HOLD&"\%1 "..\..\..\..\..\Infra\EFS\AutoFORMLaserNet
    \AutoFORMLaserNet\inputs\File Input"
```

*(Make sure that the target directory exists - in my case 2 last folders - "\inputs\File Input" - were missing...)*

Step 2.  DE.FORM.TYPE that refers to this “printer”:

T24

```
Model Bank R14          DE.FORM.TYPE SEE

    FORM.TYPE......... SYSTEM
------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. Standard landscape form
  2 PRINTER.ID........ AUTOFORMLASERNET
  3 FORM.WIDTH........ 132
  4 FORM.DEPTH........ 66
  7. 1 RPT.ATTRIBUTES. LANDCOMP
  8. 1 OPTIONS........ NHEAD
 10 CURR.NO........... 7
 11. 1 INPUTTER....... 680_INPUTTER__OFS_BROWSERTC
 12. 1 DATE.TIME...... 31 MAY 14 10:54
 13 AUTHORISER........ 680_INPUTTER_OFS_BROWSERTC
 14 CO.CODE........... GB-001-0001         Model Bank R14
 15 DEPT.CODE......... 1                   Implementation




------------------------------------------------------------------------------
19 MAR 2015 12:50:01  USER (22 APR) VLADIMIR.K        [29193,IPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

Step 3.  REPORT.CONTROL record that refers to this DE.FORM.TYPE:

T24

```
Model Bank R14          REPORT CONTROL FILE MAINTENANCE SEE

    REPORT.NAME....... ACCOUNT.STATEMENT
--------------------------------------------------------------------------------
  1. 1. 1 GB DESC..... PRE PRINTED ACCOUNT STATEMENTS
  2. 1 GB SHORT.DESC.. STATEMENTS OF ACCOUNT
  3 FORM.NAME......... SYSTEM
  8 MICROFICHE.OUTPUT. Y
 16 FICHE.SORT.NUMBER. 1
 32 REPORT.TRANSFORM.. GB0010001                              EB-MISSING.RECORD
 44 CURR.NO.......... 2
 45. 1 INPUTTER...... 76_AUTHORISER___OFS_MB.OFS.AUTH
 46. 1 DATE.TIME..... 26 NOV 07 14:15
 47 AUTHORISER....... 76_AUTHORISER_OFS_MB.OFS.AUTH
 48 CO.CODE.......... GB-001-0001         Model Bank R14
 49 DEPT.CODE........ 1                    Implementation




--------------------------------------------------------------------------------
19 MAR 2015 12:51:42  USER (22 APR) VLADIMIR.K        [29193,IPAGE 1
ACTION _
AWAITING PAGE INSTRUCTIONS
```

*(Field 32 is probably a leftover from incorrect conversion; it’s possible to edit it only in JED but it can be left as it is...)*

To test this functionality we can use ENQUIRY.REPORT application with V function:

T24

```
Model Bank R14          ENQUIRY.REPORT VERIFY

    KEY.............. ACCOUNT.STATEMENT
--------------------------------------------------------------------------------
  1. 1 GB DESCRIPTION. MB ACCOUNT STATEMENT
  2. 1 ENQUIRY........ ACCOUNT.STATEMENT   STMT.ENTRY
  3. 1. 1 SELECTION... STATEMENT.ID
  4. 1. 1 OPERAND..... EQ
  5. 1. 1 LIST........ 14613
  6. 1. 1 SORT........
  7 REPORT.CONTROL.... ACCOUNT.STATEMENT   STATEMENTS OF ACCOUNT
  8 STANDARD.HEADING..
  9. 1 ADDI
 10. 1 GB REPORT.HDR..
 11 OUTPUT.FORMAT.....
 12 RESERVED.13.......
 13 RESERVED.12.......
 14 RESERVED.11.......
 15 RESERVED.10.......
 16 RESERVED.9........
--------------------------------------------------------------------------------
```

After commit in the target directory appeared the file 17245296084382500.  To be able to overcome the space in directory name in jQL command let’s create a VOC entry for the target directory:

jsh

```
JED VOC AUTOFLNET
```

JED

```
    File VOC , Record 'AUTOFLNET'                                Insert      12:20:01
Command->
0001 F
0002 ..\..\..\..\..\Infra\EFS\AutoFORMLaserNet\AutoFORMLaserNet\inputs\File Input
0003
------------------------------- End Of Record -------------------------------
```

Then we can use this “shortcut”:

jsh

```
LIST AUTOFLNET
```

Output

```
LIST AUTOFLNET                              PAGE    1 12:21:25  19 MAR 2015

DICT AUTOFLNET

17245296084382500

 1 Records Listed
```

jsh

```
CT AUTOFLNET
```

Output

```
    17245296084382500
001
002
003
004
005
006
007
008 .
009
010
011
012
013
014
015 .
```

```
016
017
018
019   DELL COMPUTER
020
021   Dell Computer                                Account Number :    14613
022                                                Account Type :      Current
023   1 DELL WAY                                   Currency :          US Dolla
    r
024   ROUND ROCK                                   Statement Date :    22 APR 2
    014
025                                                Page :              1
026                                                Joint Holders :
027                                                From Date :         01 APR 2
    014
028                                                To Date :           22 APR 2
    014
029
030                                                Brought Forward           1
    ,744,895.85
031
...
```

# 2 How-tos

## 2.1 See record size in hashed file

jsh

```
LIST F.COMPANY *A9999
```

Output

```
  GB0010003             2904
  GB0010005             2920
  SG0010001             2925
  EU0010001             2917
  GB0010002             2912
  GB0010004             2904
  GB0010001             2976
```

And this method can show the size of a record in any jBASE file, including SQL ones:

jsh

```
LIST F.COMPANY EVAL "LEN(@RECORD)"
```

## 2.2 Wrap a long line in jBC

Method 1. Use backslash:

jBC

```
the_year = OCONV(DATE(), 'DY')
rep_line = 'The report for the year ' : the_year : ', prepared at ' \
    : TIMEDATE()
CRT rep_line
```

Method 2. If a comma (that is part of statement syntax) is at the line end:

jBC

```
days_ahead = RND(1000)
the_year = OCONV(DATE() + days_ahead,
   'DY')
CRT 'The year', days_ahead, 'days later',
   'will be', the_year
```

## 2.3 Several statements on the same line in jBC

jBC

```
the_count = 1  ;  the_count++  ;  CRT the_count
the_count++  ;* The rest of this line is a comment   ;  CRT the_count
```

Output

```
2
```

## 2.4 Comments in jBC

jBC

```
* This is a comment
! And this is a comment
REM This is also a comment
// Even this is a comment
CRT '1'    ;*    this is a comment sharing the same line with some code
CRT '2'    // yet another way to define a comment
```

## 2.5 Define a string variable in jBC

jBC

```
a_string = 'Hello'
other_string = "Dolly"
another_string = \I'm here\
```

## 2.6 Concatenate strings in jBC

jBC

```
a_string = 'Hello'
other_string = "Dolly"
another_string = \I'm here\
CRT a_string : ' ' : other_string : ", " : another_string
```

## 2.7 Put a string into single or double quotes in jBC

jBC

```
a_string = 'Hello'
CRT a_string
CRT SQUOTE(a_string)
CRT DQUOTE(a_string)
```

Output

```
Hello
'Hello'
"Hello"
```

## 2.8 Extract a substring in jBC

jBC

```
a_string = 'Hello'
* from the start
CRT a_string[1,3]
* somewhere in the middle
CRT a_string[3,2]
* to the very end
CRT DQUOTE(a_string[3,999])
* from the end
CRT a_string[2]
```

Output

```
Hel
ll
"llo"
lo
```

## 2.9 Assign new value to a substring in jBC

jBC

```
a_string = 'Hello aunt Agatha'
a_string[13] = ' broke loose'
CRT a_string                          ;* output: Hell broke loose
```

## 2.10 Get string length in jBC

jBC

```
a_string = 'Hello'
CRT LEN(a_string)                     ;* output: 5

* UTF
utf_string = 'ABCDEF' \
   : CHAR(353) : CHAR(352) : CHAR(269) : CHAR(268) : CHAR(263) \
   : CHAR(262) : CHAR(382) : CHAR(381) : CHAR(273) : CHAR(272)
CRT LEN(utf_string)                   ;* 16 characters
CRT BYTELEN(utf_string)               ;* 26 bytes

CRT OCONV(FMT(FMT(utf_string, 'MX'), '2L'), 'MCP ')
* 41 42 43 44 45 46 C5 A1 C5 A0 C4 8D C4 8C C4 87 C4 86 C5 BE C5 BD C4 91 C4 90
* e.g. C490 means "LATIN CAPITAL LETTER D WITH STROKE"

* [] takes characters, not bytes
CRT FMT(utf_string[2], 'MX')          ;* output: C491C490
CRT FMT(utf_string[9,1], 'MX')        ;* output: C48D
```

## 2.11 Repeat a string or a character many times in jBC

jBC

```
CRT STR('-', 50)
CRT STR('Bassington-', 2)[1,-2]     ;* output: Bassington-Bassington
```

## 2.12 Replace characters in a string using substitution table in jBC

Using CONVERT statement:

jBC

```
the_string = 'ABC1DEF2XYZ3321'
CONVERT '123' TO '456' IN the_string         ;* 1 -> 4, 2 -> 5, 3 -> 6
CRT the_string                               ;* ==> ABC4DEF5XYZ6654
```

Using CONVERT() function:

jBC

```
IF NOT(GETENV('JBCEMULATE', jbc_emu)) THEN
   CRT 'Emulation setting not found'
   STOP
END

the_string = 'ABC1DEF2XYZ3321'
IF jbc_emu = 'prime' THEN
   the_result = CONVERT('123', '456', the_string)
END ELSE
   the_result = CONVERT(the_string, '123', '456')
END
CRT jbc_emu, the_result                      ;* ==> prime ABC4DEF5XYZ6654
```

## 2.13 Replace substring in a string in jBC

jBC

```
* Method 1
the_string = 'ABCDEFCDYZ'
CHANGE 'CD' TO 'nnn' IN the_string
CRT the_string                                    ;* ==> ABnnnEFnnnYZ

* Method 2
CRT CHANGE('WXYZ', 'YZ', '+-=/')                  ;* ==> WX+-=/

* Method 3
CRT OCONV('Try it', 'MCC;it;that')                ;* ==> Try that
```

## 2.14 Increase/decrease a number in jBC

jBC

```
the_count = 1     ;  the_count++    ;  CRT the_count   ;* output: 2
the_count += 1    ;  CRT the_count                     ;* output: 3
the_count--       ;  CRT the_count                     ;* output: 2
the_count -= 10   ;  CRT the_count                     ;* output: -8

* and this wouldn't even compile:
the_list = 0     ;  the_list<2> = 0   ;  the_list<2> ++   ;* the_list<2> += 1 is OK
```

## 2.15 Get the absolute value in jBC

jBC

```
the_count = -10  ;  CRT ABS(the_count)               ;* output: 10
```

## 2.16 Pad a number with zeroes in jBC

jBC

```
CRT FMT(123, 'R%7')                                    ;* output: 0000123
```

## 2.17 Get division remainder in jBC

jBC

```
CRT MOD(2014, 1000)                                    ;* output: 14
```

## 2.18 Power of a number in jBC

jBC

```
CRT 2 ** 10                                            ;* output: 1024
CRT 2 ^ 10                                             ;* same as above
```

## 2.19 Precision in jBC

jBC

```
PRECISION 6
CRT 17 / 7                                             ;* output: 2.428571
PRECISION 17
CRT 17 / 7                                             ;* output: 2.42857142857142857
```

*(Default T24 PRECISION is 13 - defined in I_COMMON.)*

## 2.20 Create a dynamic array in jBC

jBC

```
* Method 1
the_array = ''
the_array<-1> = 'Field 1'
the_array<-1> = 'Field 2' :@VM: 'Field 2, value 2'
the_array<-1> = 'Field 3'
CRT OCONV(the_array, 'MCP')
* another method of getting this output:  CRT the_array 'MCP'
```

Output

```
Field 1^Field 2]Field 2, value 2^Field 3
```

jBC

```
* Method 2
the_array = ''
the_array<1> = 'Field 1'
the_array<2,1> = 'Field 2, value 1'
the_array<2,2> = 'Field 2, value 2'
the_array<3> = 'Field 3'
the_array<4> = 'Field 4'
the_array<4,1> = 'Field 4, value 1'
the_array<4,2,1> = 'Field 4, value 2, subvalue 1'
the_array<4,2,2> = 'Field 4, value 2, subvalue 2'
CRT ''

the_len = DCOUNT(the_array, @FM)

FOR i = 1 TO the_len
   CRT OCONV(the_array<i>, 'MCP')
NEXT i
```

Output

```
Field 1
Field 2, value 1]Field 2, value 2
Field 3
Field 4, value 1]Field 4, value 2, subvalue 1\Field 4, value 2, subvalue 2
```

## 2.21 Increase each element of dynamic array by a constant in jBC

jBC

```
the_array = 500 :@VM: 400 :@VM: 300 :@SM: 200 :@SM: 100 :@FM: -40
new_array = ADDS(the_array, REUSE(42))
CRT OCONV(new_array, 'MCP')                  ;* ==> 542]442]342\242\142^2
```

## 2.22 Concatenate elements of 2 dynamic arrays in jBC

jBC

```
cv_list = 'T24 local developments: '
cv_list<-1> = 'Interfaces: '
cv_list<-1> = 'T24 setup: '
cv_list<-1> = 'T24 administration: '
cv_list<-1> = 'Performance tuning: '

exp_list = 'master'
exp_list<-1> = 'advanced'
exp_list<-1> = 'beginner'
exp_list<-1> = 'advanced'
exp_list<-1> = 'master'

CRT CHANGE(CATS(cv_list, exp_list), @FM, CHAR(10))
```

Output

```
T24 local developments: master
Interfaces: advanced
T24 setup: beginner
T24 administration: advanced
Performance tuning: master
```

## 2.23 Concatenate elements of dynamic array with the same string in jBC

jBC

```
cv_list = 'T24 local developments'
cv_list<-1> = 'Interfaces'
cv_list<-1> = 'T24 setup'
cv_list<-1> = 'T24 administration'
cv_list<-1> = 'Performance tuning'

CRT CHANGE(CATS(cv_list, REUSE(' and what not')), @FM, CHAR(10))
```

Output

```
T24 local developments and what not
Interfaces and what not
T24 setup and what not
T24 administration and what not
Performance tuning and what not
```

## 2.24 Replace substring in dynamic array in jBC

jBC

```
dyn_array = the_string : @FM: the_string
CHANGE 'nnn' TO 'z' IN dyn_array
CRT OCONV(dyn_array, 'MCP')                          ;* ==> ABzEFzYZ^ABzEFzYZ
```

## 2.25 Get environment variable in jBC

jBC

```
IF NOT(GETENV('JBCEMULATE', jbc_emu)) THEN
    CRT 'Emulation setting not found'
END ELSE CRT jbc_emu                                 ;* expected output: prime
```

## 2.26 Set environment variable in jBC

jBC

```
set_non_num = 'JBASE_ERRMSG_NON_NUMERIC'
IF NOT(PUTENV(set_non_num : '=35')) THEN
   CRT 'PUTENV failed'
   STOP
END

non_num = 'A'

CRT 'See the impact of ' : set_non_num
CRT '=35: ' :
CRT non_num++

IF NOT(PUTENV(set_non_num : '=0')) THEN
   CRT 'PUTENV failed'
   STOP
END

CRT '=0: ' :
CRT non_num++
```

Output

```
See the impact of JBASE_ERRMSG_NON_NUMERIC
=35: A
=0: 1
```

## 2.27 Get number of milliseconds past midnight in jBC

jBC

```
CRT SYSTEM(12)                                    ;* sample output: 31450674.1553
```

## 2.28 “Greater-than”, “Less-than”, “Equal”, “Non-equal” notations in jBC

jbc

```
a_number = 5
IF a_number GT 3 THEN CRT a_number : ' is bigger than 3'
IF a_number LT 5 THEN CRT a_number : ' is less than 5'
IF a_number LE 5 THEN CRT a_number : ' is less or equal to 5'
* GE is 'greater or equal to'
a_string = 'ABC'
IF a_string NE 'A' THEN CRT DQUOTE(a_string) : ' is not an "A"'
IF a_string # 'B' THEN CRT DQUOTE(a_string) : ' is not a "B"'
IF a_string ! 'C' THEN CRT DQUOTE(a_string) : ' is not even a "C"'
IF a_string <> 'D' THEN CRT DQUOTE(a_string) : ' - surprisingly - neither is a "D"'
```

## 2.29 Append the existing text log or report in jBC

Use clause APPEND when you write to a sequential file with WRITESEQ:

jBC

```
out_dir = '&TEMP&'
out_file = 'events.log'
OPENSEQ out_dir, out_file TO f_out THEN NULL
WRITESEQ TIMEDATE() : '; event ' : RND(10) APPEND TO f_out ELSE
   CRT 'Write error'
   STOP
END
```

Run this program several times, then see the results:

jsh

```
CT &TEMP& events.log
```

Sample output

```
    events.log
001 07:28:08 24 FEB 2015; event 3
002 07:28:13 24 FEB 2015; event 6
003 07:28:14 24 FEB 2015; event 7
004 07:28:15 24 FEB 2015; event 8
```

## 2.30 Truncate the existing text file

jBC

```
out_dir = '&TEMP&'
out_file = 'events.log'
OPENSEQ out_dir, out_file TO f_out THEN
   WEOFSEQ f_out
END
```

## 2.31 Compile a program in one step

Program has to have “.b” extension and to be located in bnk.run

jsh

```
jcompile test.b
```

## 2.32 Compile and catalog a subroutine or a program in BP

jsh

```
BASIC -I../T24_BP ETC.BP TEST.SUB
CATALOG ETC.BP TEST.SUB
```

## 2.33 Suppress Java check etc for EB.COMPILE if you still prefer it, just compile and catalog

jsh

```
EB.COMPILE ETC.BP TEST.SUB -ct
```

## 2.34 Compile many source files at once

Use a SELECT list:

jsh

```
SELECT ETC.BP WITH @ID UNLIKE $... AND WITH @ID UNLIKE I_...
   18 Records selected

>SAVE.LIST TO-COMP
   18 record(s) saved to list 'TO-COMP'

GET.LIST TO-COMP
>BASIC -I../T24_BP ETC.BP

GET.LIST TO-COMP
>CATALOG ETC.BP
```

*(The libraries will be rebuilt by CATALOG only once.)*

## 2.35 Check some functionality without creating a program

jsh

```
LIST . EVAL "'I':FMT(41, 'R%3')" SAMPLE 1 ID-SUPP
```

Output

```
LIST . EVAL "'I':FMT(41, 'R%3')" SAMPLE 1 ID-SUPP PAGE    1 16:46:43  06 NOV 2014

"I":FMT(41, "R%3")

I041

 1 Records Listed
```

## 2.36 Get all variables in jBC

jBC

```
dump_file = 'coredump42'

rc = JBASECOREDUMP(dump_file, 0)
OSREAD all_vars FROM dump_file ELSE
   CRT 'Error reading dump file'
   STOP
END

CHANGE @FM TO CHAR(10) IN all_vars
CRT all_vars
```

Sample output

```
jBASE Core dump created at Wed Nov 19 12:01:26 2014
Program test , port 2 , process id 4056

CALL/GOSUB stack

Backtrace:
#0: jmainfunction.b:2
#1: test.b:2 -> Line 2 , Source jmainfunction.b

Backtrace log:
Program jmainfunction.b, Line 2, Stack level 0
Line 0 , Source jmainfunction.b , Level 0
>>> Program test.b, Line 2, Stack level 1
Line 2 , Source jmainfunction.b

All the defined VAR's in the program

SUBROUTINE main()
  008DFC44 : dump_file : (V) String        : 10 bytes at address 00F0A240 : coredump42
  008DFC6C : FM        : (V) Uninitialised : (UNASSIGNED)
  008DFC94 : rc        : (V) String        : 0  bytes at address 5F7F1690 :
  008DFCBC : param     : (V) Uninitialised : (UNASSIGNED)
  008DFCE4 : all_vars  : (V) Uninitialised : (UNASSIGNED)
```

## 2.37 NSELECT, XSELECT

XSELECT negates the previous select:

jsh

```
COUNT F.COMPANY
 7 Records counted

SELECT F.COMPANY SAMPLE 5
 5 Records selected

>XSELECT F.COMPANY
 2 Records selected
```

NSELECT leaves only records that are NOT in the consequent SELECT, e.g.  see which USER>COMPANY.CODE values are not @IDs of F.COMPANY:

jsh

```
SELECT F.USER SAVING UNIQUE COMPANY.CODE

 8 Records selected

>NSELECT F.COMPANY

 1 Records selected

>SAVE.LIST CMP
1 record(s) saved to list 'CMP'

CT &SAVEDLISTS& CMP
```

Output

```
    CMP
001 ALL
```

## 2.38  Delete a list in a jBC

Use the statement DELETELIST:

jBC

```
EXECUTE 'SELECT .' :@FM: 'SAVE-LIST BNK-RUN-CONTENTS' CAPTURING output
EXECUTE 'LIST &SAVEDLISTS& LIKE BNK-RUN-...'    ;* out list presents
DELETELIST 'BNK-RUN-CONTENTS'
EXECUTE 'LIST &SAVEDLISTS& LIKE BNK-RUN-...'   ;* and now it does not
```

Sample output

```
LIST &SAVEDLISTS& LIKE BNK-RUN-...         PAGE    1 21:41:06  01 DEC 2014

@ID

BNK-RUN-CONTENTS

 1 Records Listed
LIST &SAVEDLISTS& LIKE BNK-RUN-...         PAGE    1 21:41:06  01 DEC 2014

@ID

 No Records Listed
```

*(“CAPTURING output” allows to suppress the output “326 Records selected...  326 record(s) saved to list ’BNK-RUN-CONTENTS”’.)*

## 2.39 Create hashed file

jsh

```
CREATE-FILE F.SAMPLE 1 101
```

Output

```
[ 417 ] File F.SAMPLE]D created , type = J4
[ 417 ] File F.SAMPLE created , type = J4
```

## 2.40 See file statistics

jsh

```
jstat -v F.SAMPLE
```

Sample output

```
File D:\Temenos\ModelBank-R14\T24\Env\MB\bnk\bnk.run\F.SAMPLE
Type=J4 , Hash method = 5
Created at Wed Nov 19 12:06:08 2014
Groups = 101 , Frame size = 4096 bytes , Secondary Record Size = 8192 bytes
Restore re-size parameters : (none)
File size = 417792 bytes , Inode = 82385 , Device = Id 52315
Last Accessed Wed Nov 19 08:06:08 2014 , Last Modified Wed Nov 19 08:06:08 2014
Backup = YES , Log = YES , Rollback = YES , Network = NO

Record Count = 0 , Record Bytes = 0
Bytes/Record = 0 , Bytes/Group = 0
Primary   file space:   Total Frames = 101 , Total Bytes = 0
Secondary file space:   Total Frames = 0 , Total Bytes = 0
```

## 2.41 Create jBASE file for RDBMS XML storage

jsh

```
CREATE-FILE TEMP.TEMP.SW TYPE=XMLMSSQL

[ 417 ] File TEMP.TEMP.SW]D created , type = XMLMSSQL
[ 417 ] File TEMP.TEMP.SW created , type = XMLMSSQL
```

## 2.42 List users with most recent activity

jsh

```
LIST F.OS.TOKEN WITH EVAL "DATE()-TIME.2" EQ 0 USER.ID EVAL "FMT(INT((TIME()-TIME.1)),
  'R%4')" AS Sec.ago BY Sec.ago
```

## 2.43 List branches with no active users

jsh

```
SELECT F.OS.TOKEN SAVING UNIQUE COMPANY
XSELECT F.COMPANY
LIST F.COMPANY COMPANY.NAME
```

## 2.44 Get the time of generating a STMT.ENTRY up to a second

jsh

```
LIST FBNK.STMT.ENTRY DATE.TIME EVAL "OCONV(@ID[11,5],'MTS')"
```

Sample output

```
@ID..................... 170470014838222.000002
DATE.TIME.............. 1409021037
OCONV(@ID[11,5],"MTS"). 10:37:02
```

*(38222 = 10:37:02)*

Getting the same output using CONV keyword:

jsh

```
LIST FBNK.STMT.ENTRY DATE.TIME EVAL "@ID[11,5]" CONV 'MTS'
```

## 2.45 Log in to T24 automatically

Create a paragraph in VOC, e.g.:

VOC record

```
    MY.LOG
001 PA
002 ETS
003 DATA INPUTT
004 DATA 123456
005 EX
```

Put your user name to field 3, password to field 4.  Then log in automatically typing MY.LOG.

*(Note that password here is in plain text so never use it in production!)*

## 2.46 Measure time of SELECT in jQL

jsh

```
time SELECT FBNK.ACCOUNT
```

Sample output

```
 487579 Records selected

usr: 2.19   sys: 0.00   elapsed: 0m2.19s
>
```

## 2.47 Measure time of SELECT in jBC

jBC

```
EXECUTE 'SELECT -Js FBNK.ACCOUNT'
```

Sample output

```
487579 Records selected

jsh   -->usr: 2.03   sys: 0.00   elapsed: 0m2.03s
```

*(Output however comes to the screen only and can’t be captured with CAPTURING option of EXECUTE.)*

## 2.48 See which APPLICATIONs don’t have “comma VERSIONs”

jsh

```
SELECT F.PGM.FILE WITH TYPE EQ H U SAVING EVAL "@ID:','"
>SELECT F.VERSION
```

Sample output

```
...
Record 'ABBREVIATION,' is not on file.
 ** Error [ 202 ] **
Record 'PP.BALANCE.CHECK.REQUIRED,' is not on file.
 ** Error [ 202 ] **
Record 'PP.VIRTUAL.QUEUE.LIST,' is not on file.
 ** Error [ 202 ] **
Record 'LI.RISK.COLLAT.PARAM,' is not on file.
 ** Error [ 202 ] **
Record 'PP.MSGMAPPINGPARAMETER,' is not on file.
 ** Error [ 202 ] **
Record 'PW.ACTIVITY.OBJECT,' is not on file.
 ** Error [ 202 ] **
Record 'PP.SODEOD.JOBLIST.CLEAR,' is not on file.
 ** Error [ 202 ] **
Record 'PP.CHATS.DIRECTORY,' is not on file.

 1293 Records selected
```

## 2.49 See which “comma VERSIONs” have wrong number of authorisations

jsh

```
LIST F.VERSION NO.OF.AUTH WITH @ID LIKE "...','" AND NO.OF.AUTH NE 0
```

Sample output

```
@ID.................................................    NO.OF.AUTH

AA.ARR.AZ.DEPOSIT,                                     1
AA.ARR.CHANGE.PRODUCT,                                 1
AA.ARR.PAYMENT.RULES,                                  1
...
SC.PRE.DIARY,                                          1
AA.ARR.ACTIVITY.PRESENTATION,                          1
AA.SIM.ACTIVITY.RESTRICTION,                           1
AA.SIM.OFFICERS,                                       1
```

## 2.50 Get active COB job list

jsh

```
LIST F.LOCKING WITH CONTENT LIKE "'F.JOB.LIST.'..."
```

Sample output

```
@ID..... BNK/EOD.CUST.CHARGE-EB.EOD.CUST.CHARGE-1
@ID..... BNK/EOD.CUST.CHARGE-EB.EOD.CUST.CHARGE-1
KEY..... BNK/EOD.CUST.CHARGE-EB.EOD.CUST.CHARGE-1
CONTENT. F.JOB.LIST.6
```

To see more information about this job a more generic jQL request can be used:

jsh

```
LIST F.LOCKING LIKE "3X'/'0X'-'0X'-'0N" CONTENT COL.HDR 'LIST'
   EVAL "OCONV(FIELD(@ID,'-',1),'TF.BATCH;V1;1;1')" COL.HDR 'STAGE'
   EVAL "OCONV(FIELD(@ID,'-',1),'TF.BATCH;V1;6;6')" COL.HDR 'JOBS'
   EVAL "OCONV(FIELD(@ID,'-',1),'TF.BATCH;V1;12;12')" COL.HDR 'JOBSTAT'
```

## 2.51 Get total duration of all COB jobs

Example for 05.11.2014 as the last COB date:

jsh

```
LIST F.JOB.TIMES WITH EVAL "OCONV(FIELD(@ID,'-',1),'TF.BATCH;V1;1;1')" NE ''
   AND EVAL "FIELD(BATCH.DATE,@VM,1)" EQ 20141105
TOTAL EVAL "FIELD(ELAPSED.TIME,@VM,1)/3600" AS Hours DET-SUPP
```

Sample output

```
@ID.......................................................................... Hours

***                                                                          3.773

 1649 Records Listed
```

## 2.52 Get list of longest jobs

Example for 05.11.2014 as the last COB date:

jsh

```
LIST F.JOB.TIMES WITH EVAL "OCONV(FIELD(@ID,'-',1),'TF.BATCH;V1;1;1')" NE ''
   AND EVAL "FIELD(BATCH.DATE,@VM,1)" EQ 20141105 EVAL "FIELD(ELAPSED.TIME,@VM,1)/60"
   AS Minutes BY.DSND Minutes
```

Sample output

```
@ID.......................................................................... Minutes

BNK/LIMIT.REPORTS.EOD-EB.PRINT                                              22.8333
BNK/EB.CONSOL.PRINT-EOD.RE.CONSOL.PRINT                                     13.7666
BNK/LD.END.OF.DAY-LD.PROCESS.PROVISION                                      11.1333
BNK/REPORT.PRINT.SYSTEM-EB.EOD.REPORT.PRINT                                 10.7333
BNK/LD.END.OF.DAY-LD.END.OF.DAY.2                                                 6
BNK/PM.EOD2-PM.ACTIVITY.ONITE                                                5.6666
...
```

## 2.53 Compare two COB timings

Example - to compare 04.11.2014 to 05.11.2014 (which is the last COB date):

jsh

```
LIST F.JOB.TIMES WITH EVAL "OCONV(FIELD(@ID,'-',1),'TF.BATCH;V1;1;1')" NE ''
   AND EVAL "FIELD(BATCH.DATE,@VM,1)" EQ 20141105
   AND EVAL "FIELD(BATCH.DATE,@VM,2)" EQ 20141104
EVAL "(FIELD(ELAPSED.TIME,@VM,1)-FIELD(ELAPSED.TIME,@VM,2))/60" AS DIFF BY.DSND DIFF
```

Sample output

```
@ID.......................................................................... DIFF...

BNK/SYSTEM.END.OF.DAY3-IC.COB                                                  1.05
BNK/LD.END.OF.DAY-LD.END.OF.DAY.2                                            1.0333
BNK/RE.BUILD.SLC-RE.RECALC.REP.WORK                                            0.75
BNK/PM.EOD2-PM.CLEAR.DPC                                                     0.5333
BNK/PD.END.OF.DAY-PD.EOD.CONTRACT                                               0.4
BNK/FILE.TIDY.UP-EOD.CLEAR.PROTOCOL                                          0.3833
...
```

## 2.54 Run jsh command, jQL query or a program on the server using jRemote

t24get.java source code

```
import com.jbase.jremote.DefaultJConnectionFactory;
import com.jbase.jremote.JConnection;
import com.jbase.jremote.JConnectionFactory;
import com.jbase.jremote.JDynArray;
import com.jbase.jremote.JRemoteException;
import com.jbase.jremote.JExecuteResults;
import java.util.Properties;
import java.io.ByteArrayOutputStream;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.io.UnsupportedEncodingException;

public class t24get {
   private JConnectionFactory factory = null;
   public t24get(JConnectionFactory f) {
      factory = f;
   }

public void perform(String cmd, String host) {
   try {
      Properties prop = new Properties();
      prop.setProperty("allow input", "true");
      JConnection c = factory.getConnection(null, null, prop);
      ByteArrayOutputStream bos = new ByteArrayOutputStream();
      Writer writer = new OutputStreamWriter(bos, "UTF-8");
      c.setTerminalOutputWriter(writer);
      System.out.println(host + ": " + cmd);
      JExecuteResults results = c.execute(cmd);

      if (results != null) {
         JDynArray resarray = new JDynArray();
         resarray = results.getCapturingVar();
         int nattr = resarray.getNumberOfAttributes();
         //
         for(int a = 1; a <= nattr; a++) {
            System.out.println(resarray.get(a));
         }
      }

      c.close();

   } catch (UnsupportedEncodingException e) {
      System.out.println("Error creating OutputStreamWriter.");
   } catch (JRemoteException e) {
      System.out.println(e);
   } catch (Exception e) {
      System.out.println("Unknown error.");
   }
}

public static String strJoin(String[] aArr, String sSep) {
    StringBuilder sbStr = new StringBuilder();
    for (int i = 2, il = aArr.length; i < il; i++) {
```

```
        if (i > 2)
            sbStr.append(sSep);
        sbStr.append(aArr[i]);
    }
    return sbStr.toString();
}

public static void main(String[] args) {
   DefaultJConnectionFactory factory = new
   DefaultJConnectionFactory();

// mandatory host
   String host = args[0];
   factory.setHost(host);

// mandatory port
   int port = Integer.valueOf(args[1]);
   factory.setPort(port);
   //
   String extcmd = "";
   if (args != null && args.length > 2) {
      extcmd = strJoin(args, " ");
   } else {
      extcmd = "jdiag";
   }

   t24get example = new t24get(factory);
       example.perform(extcmd, host);
   }
}
```

Java compilation

```
set JAVA_HOME=D:\Temenos\ModelBank-R14\Infra\Java
set PATH=%PATH%;%JAVA_HOME%\bin
set TAFC_HOME=D:\Temenos\ModelBank-R14\T24\Programs\TAFC
set CLASSPATH=%TAFC_HOME%\java\lib\jremote.jar;%CLASSPATH%
javac t24get.java
```

Run

```
set JAVA_HOME=D:\Temenos\ModelBank-R14\Infra\Java
set PATH=%PATH%;%JAVA_HOME%\bin
set TAFC_HOME=D:\Temenos\ModelBank-R14\T24\Programs\TAFC
set CLASSPATH=%TAFC_HOME%\java\lib\jremote.jar;%CLASSPATH%
java t24get 127.0.0.1 20014 "LIST F.SPF"
```

## 2.55 Call jBC subroutine on the server using jRemote

In this example we'll call a jBC subroutine with one parameter.

t24callsub.java source code

```
//
import com.jbase.jremote.DefaultJConnectionFactory;
import com.jbase.jremote.JConnection;
import com.jbase.jremote.JConnectionFactory;
import com.jbase.jremote.JDynArray;
import com.jbase.jremote.JRemoteException;
import com.jbase.jremote.JExecuteResults;
import com.jbase.jremote.JSubroutineParameters;
import java.util.Properties;
import java.io.ByteArrayOutputStream;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.io.UnsupportedEncodingException;

public class t24callsub {
   private JConnectionFactory factory = null;
   public t24callsub(JConnectionFactory f) {
      factory = f;
   }

public void perform(String subr, String cmd, String host) {
   try {
      Properties prop = new Properties();
      prop.setProperty("allow input", "true");
      JConnection c = factory.getConnection(null, null, prop);
      ByteArrayOutputStream bos = new ByteArrayOutputStream();
      Writer writer = new OutputStreamWriter(bos, "UTF-8");
      c.setTerminalOutputWriter(writer);
      System.out.println("* " + host + ": CALL " + subr + '(' + cmd + ')');

      JSubroutineParameters subroutineParams = new JSubroutineParameters();
      subroutineParams.add(new JDynArray(cmd));

      subroutineParams = c.call(subr, subroutineParams);
      if (subroutineParams != null) {
         String res_all = subroutineParams.toString();
         int res_length = res_all.length();
// remove "[<1>"... "]":
         System.out.println(res_all.substring(4, res_length-1));
      }

      c.close();

   } catch (UnsupportedEncodingException e) {
      System.out.println("Error creating OutputStreamWriter.");
   } catch (JRemoteException e) {
      System.out.println(e);
      // e.printStackTrace(); // it goes to stderr
   } catch (Exception e) {
      System.out.println("Unknown error.");
   }
}

public static String strJoin(String[] aArr, String sSep) {
    StringBuilder sbStr = new StringBuilder();
    for (int i = 3, il = aArr.length; i < il; i++) {
        if (i > 3)
```

```
            sbStr.append(sSep);
        sbStr.append(aArr[i]);
    }
    return sbStr.toString();
}

public static void main(String[] args) {
   DefaultJConnectionFactory factory = new
   DefaultJConnectionFactory();
// mandatory host
   String host = args[0];
   factory.setHost(host);

// mandatory port
   int port = Integer.valueOf(args[1]);
   factory.setPort(port);

// mandatory subroutine name
   String subr = args[2];

   String extcmd = "";

   if (args != null && args.length > 3) {
      extcmd = strJoin(args, " ");
      t24callsub example = new t24callsub(factory);
      example.perform(subr, extcmd, host);
   } else {
      System.out.println("Wrong number of parameters.");
      System.out.println('(' + extcmd + ')');
   }

   }
}
```

Java compilation

```
set JAVA_HOME=D:\Temenos\ModelBank-R14\Infra\Java
set PATH=%PATH%;%JAVA_HOME%\bin
set TAFC_HOME=D:\Temenos\ModelBank-R14\T24\Programs\TAFC
set CLASSPATH=%TAFC_HOME%\java\lib\jremote.jar;%CLASSPATH%
javac t24callsub.java
```

Run

```
set JAVA_HOME=D:\Temenos\ModelBank-R14\Infra\Java
set PATH=%PATH%;%JAVA_HOME%\bin
set TAFC_HOME=D:\Temenos\ModelBank-R14\T24\Programs\TAFC
set CLASSPATH=%TAFC_HOME%\java\lib\jremote.jar;%CLASSPATH%
java t24callsub 127.0.0.1 20014 ALLOCATE.UNIQUE.TIME 1
```

Output

```
* 127.0.0.1: CALL ALLOCATE.UNIQUE.TIME(1)
0000023638.00
```

## 2.56 Process OFS message on the server using jRemote

We'll use java class from the previous chapter to call the following jBC subroutine:

jBC

```
// Process OFS via jremote
SUBROUTINE kzm_jremofs(param)
$INSERT I_COMMON
$INSERT I_EQUATE

   request = param
   result = ''

   IF PUTENV('OFS_SOURCE=TAABS') THEN NULL

   CALL JF.INITIALISE.CONNECTION
   CALL OFS.BULK.MANAGER(request, result, '')

   CHANGE ',' TO ',' : CHAR(10) IN result  ;* each field on a new line
   param = result

   RETURN
END
```

cmd file to run java class

```
set JAVA_HOME=D:\Temenos\ModelBank-R14\Infra\Java
set PATH=%PATH%;%JAVA_HOME%\bin
set TAFC_HOME=D:\Temenos\ModelBank-R14\T24\Programs\TAFC
set CLASSPATH=%TAFC_HOME%\java\lib\jremote.jar;%CLASSPATH%
java t24callsub 127.0.0.1 20014 kzm_jremofs ^
"AB,/I/PROCESS//0,INPUTT/123456,EDITSPF,ORIGINAL.TEXT::=SPF I SYSTEM"
```

jRemote addresses jbase agent directly (JBoss isn't required); besides OFS reply from java we can see the trace of our call in jbase agent window:

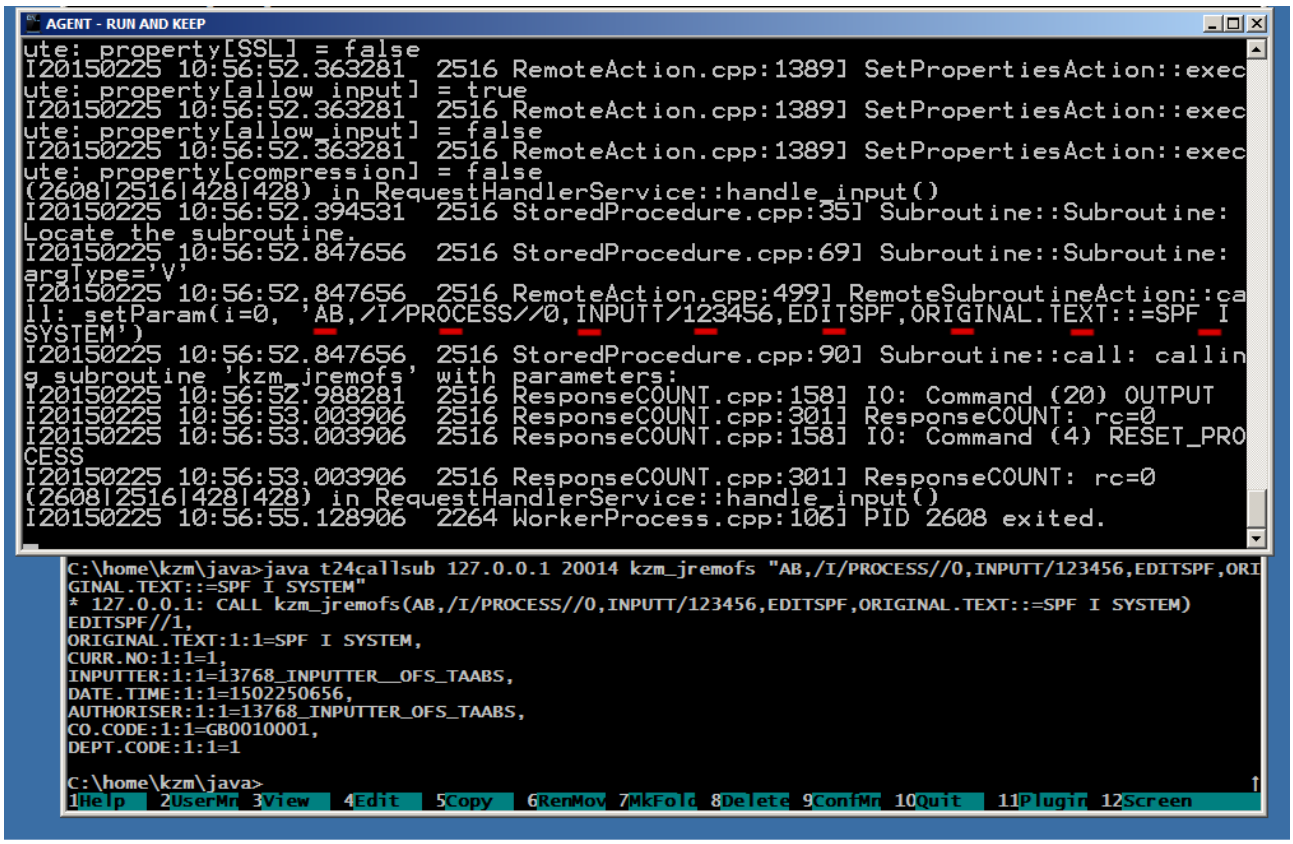


**Figure 56: jRemote call - OFS.**

Run our train enquiry TRAIN.ENQ via this interface:

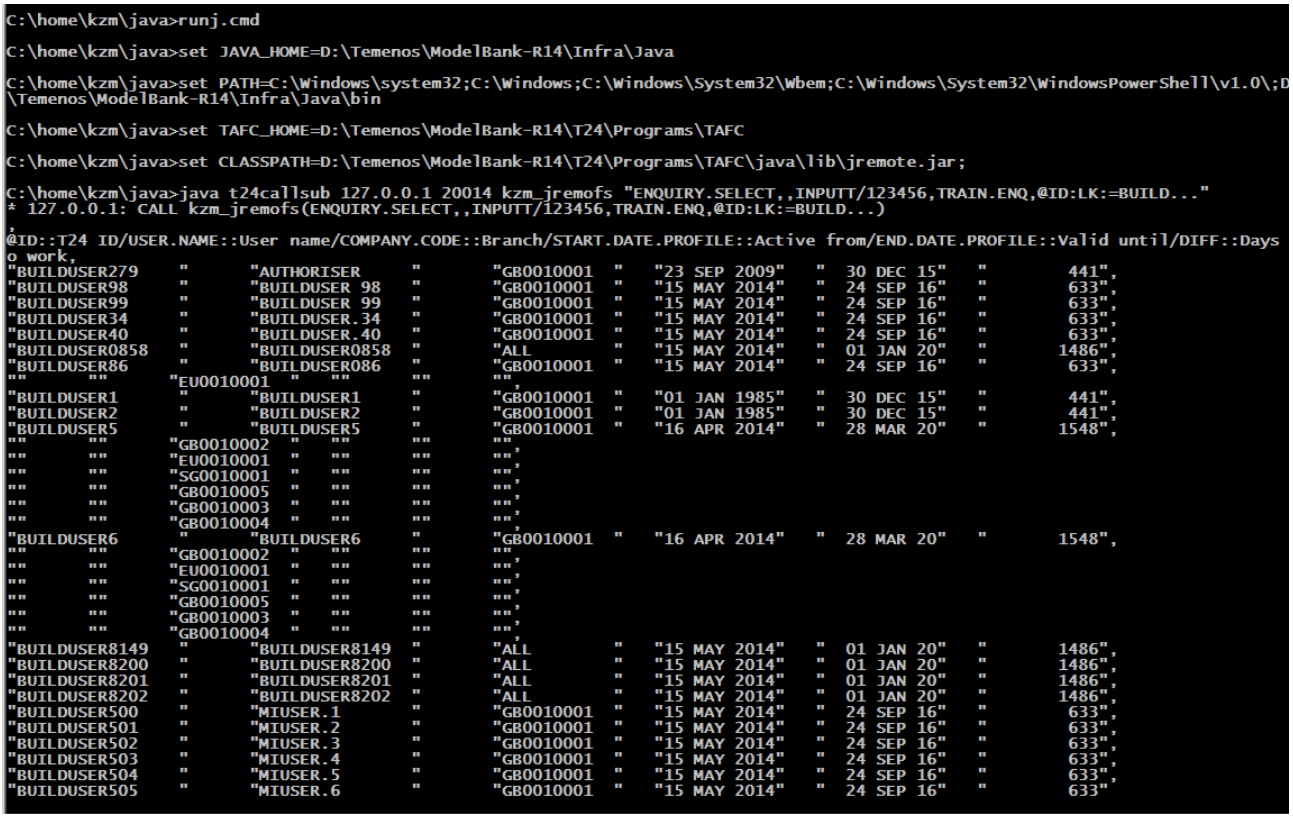


**Figure 57: jRemote call - OFS enquiry.**

## 2.57 Get the latest records from T24 protocol directly from SQL DB (MSSQL)

SQL

```
select top(5000) RECID,
   XMLRECORD.value('(/row//c4/node())[1]', 'nvarchar(max)') [timer],
   XMLRECORD.value('(/row//c8/node())[1]', 'nvarchar(max)') [t24user],
   XMLRECORD.value('(/row//c7/node())[1]', 'nvarchar(max)') [branch],
   XMLRECORD.value('(/row//c9/node())[1]', 'nvarchar(max)') [application],
   XMLRECORD.value('(/row//c11/node())[1]', 'nvarchar(max)') [app_id],
   XMLRECORD.value('(/row//c12/node())[1]', 'nvarchar(max)') [remark]
from F_PROTOCOL
   where RECID like '20141104%'
order by timer desc
```

## 2.58 Avoid the error “Unable to initialise DriverData Structure” (MSSQL)

If T24 environment was moved elsewhere and MS SQL DB is no more accessible, an attempt to run jsh or config.xmlmssql fails with the following error:

jsh failure

```
drvOpen: *** ERROR *** Unable to initialise DriverData Structure,
check ./XMLdriver.log for more information.
```

To cure that temporarily remove “%DRIVER_HOME%\lib” from JBCOBJECTLIST environment variable and instead of running jsh run cmd.exe, then correct the DB config using config.xmlmssql, then revert to initial settings in remote.cmd.

## 2.59 See what T24 war is deployed at JBoss and in which deploy directory it is

Step 1. Go to JBoss main page and click the link “JBoss Web Console”. See running config, e.g.:

Page contents

```
Running config: 'default'
```

...so we have this war in (JBoss)\server\default\deploy. The name (it’s not necessarily BrowserWeb): go back to JBoss main page, then click “JMX Console”. Search “server/default/deploy” there; sooner or later you’ll find something like:

Page contents

```
id="vfsfile:/C:/T24/jboss-5.1.0.GA/server/default/deploy/T24TEST.war/"
```

*(Deployments may be done in other “server” folders, e.g. /server/all/deploy.)*

## 2.60 Get rid of annoying beep in jsh

If you type some command in jshell, then use backspace to erase it more than necessary, a beep is issued. If you find it annoying, it’s no need to hush your PC sound at all.

First of all, list your TERM setting:

jsh

```
echo %TERM%
```

Output

```
VT100
```

*(Might be something else, like ntcon, vt220 etc.)*

Edit the file in TAFC directory:

jsh

```
JED D:\Temenos\ModelBank-R14\T24\Programs\TAFC\src jbase_nt.ti
```

JED

```
File D:\Temenos\ModelBank-R14\T24\Programs\TAFC\src        Insertse_nt.07:12:58
Command->
0001 #
0002 # Copyright (c) 2014 TEMENOS HOLDINGS NV. All rights reserved
0003 #
0004 # Various Terminal definitions
0005 #
0006 #.ntcon - Windows Console
0007 #.vt100/vt220.- DEC vt terminals
0008 #.wy50..- wyse 50
0009 #.addsvp..- Add viewpoint
...
```

Find settings for vt100 (in lowercase):

JED

```
String 'vt100' found                                       Insert      07:16:00
Command-> /vt100
0001 #
0002 # Copyright (c) 2014 TEMENOS HOLDINGS NV. All rights reserved
0003 #
0004 # Various Terminal definitions
0005 #
0006 #.ntcon - Windows Console
0007 #.vt100/vt220.- DEC vt terminals
...
```

That’s a comment...  Go down one line, press Esc, then /

JED

```
String 'vt100' found                                       Insert      07:17:12
Command->
0048 vt100|vt100-am|dec vt100 (w/advanced video),
0049 .am, mir, msgr, xenl, xon,
0050 .cols#80, it#8, lines#24, vt#3,
0051 .acsc=aaffggjjkkllmmnnooppqqrrssttuuvvwwxxyyzz{{||}}˜,
0052 .bel=^G, blink=\E[5m$<2>, bold=\E[1m$<2>,
0053 .clear=\E[H\E[J$<50>, cr=^M, csr=\E[%i%p1%d;%p2%dr,
...
```

Here we are...  remove setting for “bel=” at the line 52 (don’t touch the dot at position 1 - it’s a TAB actually).

Result:

JED

```
Command->
0048 vt100|vt100-am|dec vt100 (w/advanced video),
0049 .am, mir, msgr, xenl, xon,
0050 .cols#80, it#8, lines#24, vt#3,
0051 .acsc=aaffggjjkkllmmnnooppqqrrssttuuvvwwxxyyzz{{||}}~,
0052 .blink=\E[5m$<2>, bold=\E[1m$<2>,
0053 .clear=\E[H\E[J$<50>, cr=^M, csr=\E[%i%p1%d;%p2%dr,
...
```

Save the file, then “compile” it:

jsh

```
jtic D:\Temenos\ModelBank-R14\T24\Programs\TAFC\src\jbase_nt.ti
```

That’s it.

## 2.61 Get rid of annoying beep in T24 Classic

Errors in T24 classic mode are also beepy, like in the situation on the screen below:

T24

```
Model Bank R14          SELECT APPLICATION


------------------------------------------------------------------------------









-- LAST SIGN.ON, DATE: 25 FEB 2015     TIME: 07:06        ATTEMPTS: 0 --------
25 FEB 2015 07:23:39  USER (22 APR) VLADIMIR.K        [27624,IN]
ACTION EDWDE                                            APPLICATION MISSING
AWAITING APPLICATION
```

To get rid of it edit TERMINAL application:

T24

```
Model Bank R14          TERMINAL, INPUT

    TERMINAL.......... EBS-JBASE
--------------------------------------------------------------------------------
 17. 1 INIT.SEQUENCE.. <027>&oR
 17. 2 INIT.SEQUENCE.. <027>[?67h
 18. 1 FUNCTION.KEYS.. <27>&f2a1k6d1LRETURN<21>
 18. 2 FUNCTION.KEYS.. <27>&f2a2k4d1LBACK<2>
 18. 3 FUNCTION.KEYS.. <27>&f2a3k7d1LFORWARD<6>
 18. 4 FUNCTION.KEYS.. <27>&f2a4k4d1LLAST<5>
 18. 5 FUNCTION.KEYS.. <27>&f2a5k5d1LENTER<22>
 18. 6 FUNCTION.KEYS.. <27>&f2a6k4d1LNEXT<23>
 18. 7 FUNCTION.KEYS.. <27>&f2a7k4d1LLEFT<29>
 18. 8 FUNCTION.KEYS.. <27>&f2a8k5d1LRIGHT<30>
 19. 1 INIT.EXECUTES..
 20 BELL.............. <007>                          <===== make this field blank
 21 RESET............. <027>[0;1m
 22 CURSOR.BACK.......
 23 PRINT............. <027>[D
 24 PRINT.ECHO........ <027>[C
--------------------------------------------------------------------------------
25 FEB 2015 07:27:03  USER (22 APR) VLADIMIR.K        [8242,INPAGE 2   >>>5>>>
ACTION _
AWAITING PAGE INSTRUCTIONS
```

## 2.62 See C code that is generated by jBASE compiler

jsh

```
COPY FROM ETC.BP SUBR.0,SUBR.0.b
jcompile -S ETC.BP\SUBR.0.b
CT ETC.BP SUBR.0.c SUBR.0.j
```

*(jcompile needs file extension ".b" to recognize jBC source code file.)*

Output

```
   SUBR.0.c
001 /*
002 ** This program was produced by the jBASE system version 14.0
003 */
004
005 #define PRIME_FLAG 1
006 #include.<jsystem.h>./* System globals etc.*/
007 #include."SUBR.0.j"
008
009 DEFINED_MAIN
010 #line 1 "ETC.BP\SUBR.0.b"
011 JBASE_EXPORT.int.JBC_SUBR_2E0(dp, ActualFlags)
012
013
```

```
014 #line 1 "ETC.BP\SUBR.0.b"
015 struct jBASEDataAreas.*dp;
016 char.* ActualFlags;
017
018 #line 1 "ETC.BP\SUBR.0.b"
019 {
020 SUBROUTINE_DATA(0)
021 COMMON_VARIABLES
022 SUBROUTINE_INIT(dp, &StackData, ConstantText , &debug_flag_pointer , &debu
    g_line_pointer , GlobalBasicVars, (struct StaticLevelData *)&StaticData ,
...
...
...
092 RETURN_V;
093 #line 15 "ETC.BP\SUBR.0.b"
094
095
096
097 _jl_function_end:
098 SUBROUTINE_END()
099 RETURN_SWITCH
100 PROGRAM_END

    SUBR.0.j
001 /*
002 ** This program was produced by the jBASE system version 14.0
003 */
004
005 /* Cheader directives
006  */
007
008 #pragma warning(disable: 4116 4129)
...
...
...
882 RELOAD_COMMON( (char*)GlobalCommonNameASCIIEBCIDIC, &GlobalCommonVarsASCII
    EBCIDIC , 1 , &GlobalCommonVarsKeyASCIIEBCIDIC);.\
883 RELOAD_COMMON( (char*)GlobalCommonNameFNKEYS, &GlobalCommonVarsFNKEYS , 16
     , &GlobalCommonVarsKeyFNKEYS);.\
884
885
886 #define.DEFINED_MAIN
```

## 2.63 Check if variable is assigned (jBC)

jBC

```
CRT ASSIGNED(the_yesno)    ;* 0
the_yesno = 'NO'
CRT ASSIGNED(the_yesno)    ;* 1
```

## 2.64 Check how jBC variable is stored internally

jBC

```
a_string = 'Year 2015'
a_num = a_string[4]
DEBUG
a_num += 0
CRT a_num
```

Debugger

```
DEBUG statement seen
0003     DEBUG
jBASE debugger->V -v a_num
  00000000005E6320 : a_num : (V) String  : 4 bytes at address 00000000007C79A0 : 2015
jBASE debugger->S
0004     a_num += 0
jBASE debugger->S
0005     CRT a_num
jBASE debugger->V -v a_num
  00000000005E6320 : a_num                    : (V) Scaled (4) float     : 2015
```

## 2.65 Repeat commands issued earlier in T24 Classic

Commands that user types at “AWAITING APPLICATION” prompt are stored in the global variable - in the first field; in the second one there are selection criteria that user enters in enquiries. When user logs out of T24 and logs in again, this global variable is initialized again.

To use it we can write a mainline T24 subroutine:

jBC

```
   SUBROUTINE CMD.HIST
* T24 mainline routine
$INSERT I_COMMON
$INSERT I_EQUATE

   the_stack = CMD$STACK<1>
   the_len = DCOUNT(the_stack, VM)

   FOR i = 1 TO 10  ;  CRT ''  ;  NEXT i

   FOR i = 1 TO the_len
      CRT i, the_stack<1,i>
   NEXT i

   INPUT the_num
   IF the_num EQ '' THEN RETURN

   CALL EB.SET.NEXT.TASK(the_stack<1,the_num>)

   RETURN
END
```

Usage example:  log in to T24, type several commands at “AWAITING APPLICATION” prompt, then launch the subroutine CMD.HIST:

T24

```
 ------------------------------------------------------------------------------
 26 FEB 2015 08:22:29  USER (22 APR) VLADIMIR.K        [16969,IN]
 ACTION









1       CMD.HIST
2       SPF S SYSTEM
3       AC L
4       PGM.FILE, I CMD.HIST
```

Select option 2...

T24

```
 Model Bank R14          SPF SEE

     SYSTEM SPEC....... SYSTEM
 ------------------------------------------------------------------------------
   1 RUN.DATE.......... 28 MAR 2014
   2 SITE.NAME......... Model Bank R14
   3 OP.MODE........... O
   4. 1 OP.CONSOLE..... OFF
   5. 1 MAIN.ACCOUNT... ../bnk.data
   8 CURRENT.RELEASE... R14
   9 HIST.LIFE......... 1
  11 CACHE.EXPIRY...... 0
  12 ENQ.PAGE.LIMIT.... 200
  14 SYS.BACKUP.MODE... TAPE
  15 HOLD.BATCH.OUTPUT. Y
  16 MICROFICHE.OUTPUT. N
  17 REPORT.RETENTION.. 5
  19 DATA.ACC.NAME..... ../bnk.data
  20 RUN.ACC.NAME...... ../bnk.run
  21 DICT.ACC.NAME..... ../bnk.dict
 ------------------------------------------------------------------------------
 26 FEB 2015 08:23:14  USER (22 APR) VLADIMIR.K        [16969,IPAGE 1   >>>9>>>
 ACTION
 AWAITING PAGE INSTRUCTIONS
```

Try something else; launch CMD.HIST again:

T24

```
1       CMD.HIST
2       DE.O.HEADER L
3       AB, I QWERTY
4       CU L L
5       CMD.HIST
6       SPF S SYSTEM
7       AC L
8       PGM.FILE, I CMD.HIST
```

*(Code can be improved not to show the duplicate entries...)*

## 2.66 See hexadecimal representation of contents in JED

Use HEX command (acts as a toggle):

jsh

```
JED F.USER INPUTTER
```

JED

```
    File F.USER , Record 'INPUTTER'                              Insert      13:08:53
Command->
0001 INPUTTER
0002 INPUTT
0003 INT
0004 1
0005 GB0010001]GB0010002]EU0010001]SG0010001]GB0010005]GB0010003]GB0010004
0006 1
0007 20150601M0601
0008 20140515
...
```

Esc, enter HEX command:

JED

```
File F.USER , Record 'INPUTTER'                              Insert      13:09:25
Command->
0001 494E505554544552
0002 494E50555454
0003 494E54
0004 31
0005 474230303130303031FD474230303130303032FD455530303130303031FD53473030313030
0006 31
0007 323031353036303 14D30363031
0008 3230313430353135
...
```

*(We can see now value marks (FD) in the field 5; editing in this mode is also possible.)*

## 2.67 Get the command line of jBC program

jBC program PROG0

```
param_qty = DCOUNT(SYSTEM(1000), @FM) - 1
CRT param_qty, 'parameter(s) detected'
FOR i = 1 TO param_qty
   CRT 'Parameter', i, 'is', SENTENCE(i)
NEXT i
```

Run:

jsh

```
PROG0
PROG0 123
PROG0 123 456
PROG0 "123 456"
```

Output

```
0   parameter(s) detected

1   parameter(s) detected
Parameter   1   is   123

2   parameter(s) detected
Parameter   1   is   123
Parameter   2   is   456

1   parameter(s) detected
Parameter   1   is   "123 456"
```

That’s all folks!
Tools used: LaTeX, Python, SciTE editor and cat Masya. Enjoy!