Basic Version Routines

TEMENOS EDUCATION CENTRE

NOTICE

These training materials are the copyrighted work of Temenos Headquarters SA and other companies in the TEMENOS group of companies (The Copyright Owner). The training materials contain protected logos, graphics and images. Use of the training materials is restricted solely for use by licensed end users, partners and employees. Any un-licensed reproduction by any means, redistribution, editing, transformation, publishing, distribution, or public demonstration of the training materials whether for commercial or personal gain is expressly prohibited by law, and may result in severe civil and criminal penalties. Violators will be prosecuted to the maximum extent possible. Such training materials shall not be represented, extracted into or included in part, or in whole, as part of any other training documentation without the express permission of the Copyright Owner, which must given in writing by an authorised agent of the Copyright Owner to be valid. Where such permission is given a clear and prominent notice must be displayed on any and all documentation accrediting the Copyright Owner with having copyright over the materials. End-user licenses will in no event contain permissions extending the use of these training materials to third parties for commercial training purposes.

Without limiting the foregoing, copying or reproduction of the training materials in part or in whole to any other sever or location for further reproduction or redistribution is expressly prohibited, unless such reproduction is expressly licensed by the Copyright Owner.

Copyright © 2010 Temenos Headquarters SA















Objectives



After completing this learning unit, you will be able to:

- Explain the need for version routines
- Create different version routine

Versions in T24



VERSION is used to customize an application in T24

 Routines that are attached to the VERSION application are called Version Routines

Common Variables in Version Routines



What values do version routines work with?

Common variables used are defined in I_COMMON

Different common variables initialized at different points in time



- Contains the ID of the currently opened record in any application
- Dynamic array
- When the record is closed or committed or authorized the value in this variable gets cleared

R.NEW



- Holds the currently opened record in any application
- Dimensioned array
- Defined in I_COMMON as DIM R.NEW(C\$SYSDIM)
- C\$SYSDIM also a common variable equated to 500

ID.NEW and R.NEW



jBASE debugger->V ID.NEW COMMON variables ID.NEW : 111183

jBASE debugger->V R.NEW COMMON variables R.NEW(O) R.NEW(1) : CHAIT : CHAITU R.NEW(2) : CHAITANYA R.NEW(3)R.NEW(4) R.NEW(5) : STERLING R.NEW(6) : NUNGAMBAKKAM : INDIA R.NEW(7) R.NEW(8) R.NEW(9) R.NEW(10) : 1 R.NEW(11) : 100724 R.NEW(12) : 10 R.NEW(13) R.NEW(496) R.NEW(497) R.NEW(498) R.NEW(499) R.NEW (500)



Mnemonic	TEST	
GB Short Name	CHAIT TEST	
GB Name 1	CHAITANYA TEST	
GB Street	STERLING ROAD	
GB Address.1	NUNGAMBAKKAM	
GB Town Country	CHENNAI	
GB Country	INDIA	
Sector	1000	***IndividualsSS
Account Officer	1	Implementation
Industry	1000	Private Person (Name)
Target	1	Private Client - Standard
Nationality	IN	India
Customer Status	1	Private Client - Standard
Residence	IN	India
Language	1	English
Company Book	GB-001-0001	R8 MODEL BANK
Cls Cparty	NO	
Override.1	Have you received In	troductory Document/CUS*100 from 111178 RECEIVE
Record Status	INAU	INPUT Unauthorised
Curr No	1	
Inputter.1	17_CHAITANYA	DFS_BROWSERTC
Date Time.1	25 SEP 08 14:37	
Co Code	GB-001-0001	R8 MODEL BANK
Dept Code	1	

ID.NEW.LAST



- Holds the ID of the unauthorized record if any for the currently opened record
- Dynamic array
- The ID of the live and unauthorized record will be the same

R.NEW.LAST



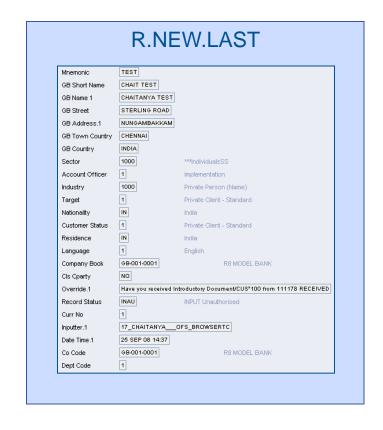
- Holds the actual unauthorized record if any for the currently opened ID
- Dimensioned array
- Defined in I_COMMON as DIM R.NEW.LAST(C\$SYSDIM)



ID.NEW









- Holds the ID of the authorized record if exists for the currently opened record
- Dynamic array
- Enables to check whether the currently opened record has a authorized record

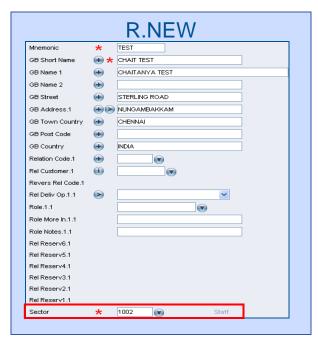
R.OLD



- Holds the actual authorized record of the currently opened record ID
- Dimensioned array
- Defined in I_COMMON as DIM R.OLD(C\$SYSDIM)



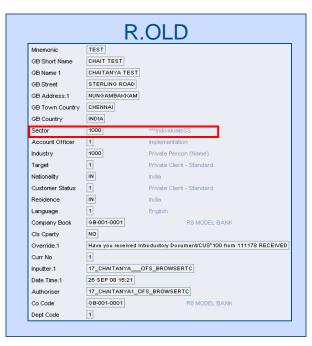




ID.NEW.LAST



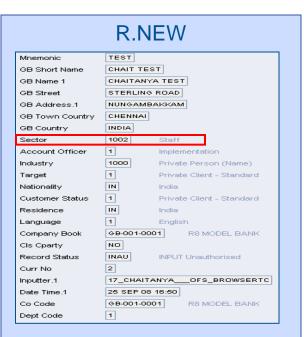




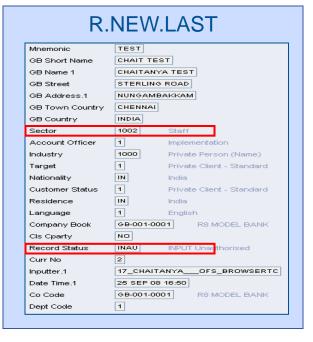
ID.NEW, R.NEW, ID.NEW.LAST, R.NEW.LAST, ID.OLD and R.OLD

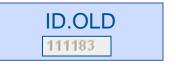














Validation Routine



- Perform field level validation
- Default values into fields
- Routine name must be specified in the field VALIDATION.FLD
- Invoked on committing the record
- Can also be invoked once tabbed out of the field if it is a hot field



Create a version for the FUNDS.TRANSFER application with the following fields

TRANSACTION.TYPE

DEBIT.ACCT.NO

DEBIT.CURRENCY

DEBIT.AMOUNT

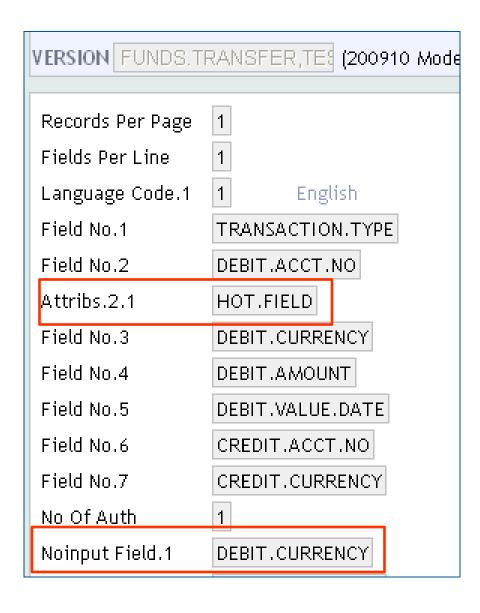
DEBIT.VALUE.DATE

CREDIT.ACCT.NO

CREDIT.CURRENCY

- Set the field DEBIT.ACCT.NO as hot field
- Set the field DEBIT.CURRENCY as NOINPUT field

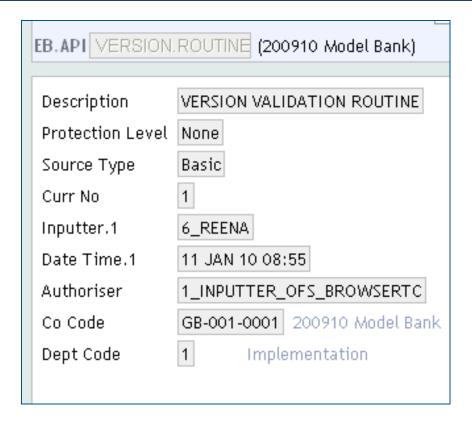


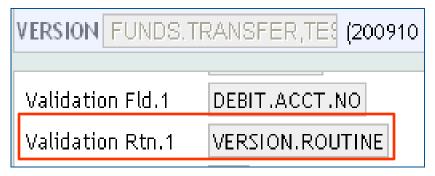




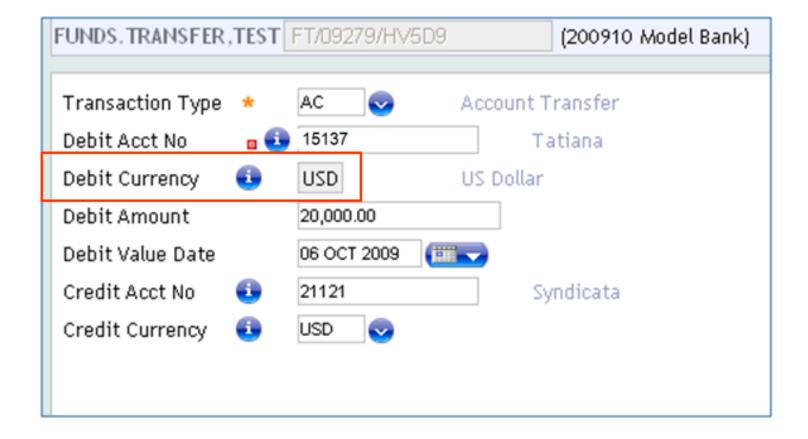
```
*Version routine to populate the debit currency
    SUBROUTINE VERSION. ROUTINE
    $INSERT I COMMON
    $INSERT I EQUATE
    $INSERT I F.ACCOUNT
    $INSERT I F.FUNDS.TRANSFER
        Y.ACCT.NO = R.NEW(FT.DEBIT.ACCT.NO) ; Y.ACCT.CUR = ''
        FN.ACCOUNT = 'F.ACCOUNT'; F.ACCOUNT = ''
        CALL OPF (FN.ACCOUNT, F.ACCOUNT)
        CALL F.READ (FN.ACCOUNT, Y.ACCT.NO, R.ACCT.REC, F.ACCOUNT, ERR1)
        Y.ACCT.CUR = R.ACCT.REC<AC.CURRENCY>
        R.NEW(FT.DEBIT.CURRENCY) = Y.ACCT.CUR
        CALL REBUILD. SCREEN
    END
    RETURN
END
```











The Banking Software Company



```
0011 DEBUG
iBASE debugger->C
DEBUG statement seen
0011 DEBUG
jBASE debugger->S
0012
            Y.ACCT.NO = R.NEW(FT.DEBIT.ACCT.NO) ; Y.ACCT.CUR = ''
iBASE debugger->S
0013
             FN.ACCOUNT = 'F.ACCOUNT' ; F.ACCOUNT = ''
iBASE debugger->V Y.ACCT.NO
  Y.ACCT.NO
                            : 15137
jBASE debugger->S
             CALL OPF(FN.ACCOUNT, F.ACCOUNT)
0014
jBASE debugger->S
0015
             CALL F.READ(FN.ACCOUNT, Y.ACCT.NO, R.ACCT.REC, F.ACCOUNT, ERR1)
BASE debugger->V FN.ACCOUNT
  FN.ACCOUNT
                            : FBNK.ACCOUNT
jBASE debugger->S
0016
            Y.ACCT.CUR = R.ACCT.REC<AC.CURRENCY>
jBASE debugger->S
0017
             R.NEW(FT.DEBIT.CURRENCY) = Y.ACCT.CUR
BASE debugger->V Y.ACCT.CUR
  Y.ACCT.CUR
                            : USD
```





D:\SignOn\jboss-4.2.3.GA\server\default\conf

```
<category name="tcs">
  <priority value="DEBUG"}>
  <appender-ref ref="CONSOLE"}>
</category>
```

HOT.FIELD

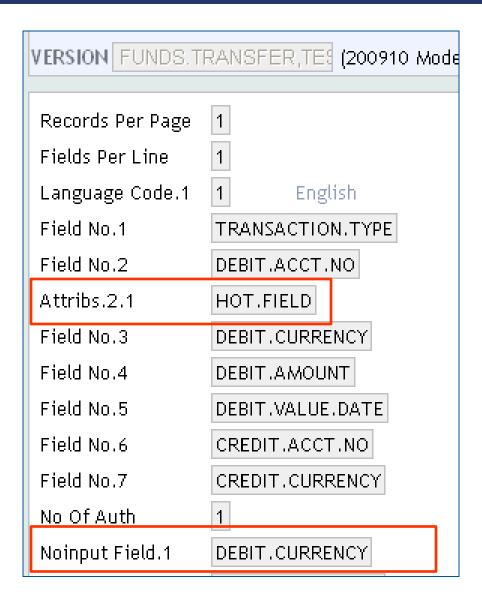


- Set the field ATTRIBS in VERSION to HOT.FIELD
- Hot fields are shown with next to them.
- Automatically sends a validation request to the server on changing the value



- COMI holds the last input field value
- COMI is a Dynamic Array

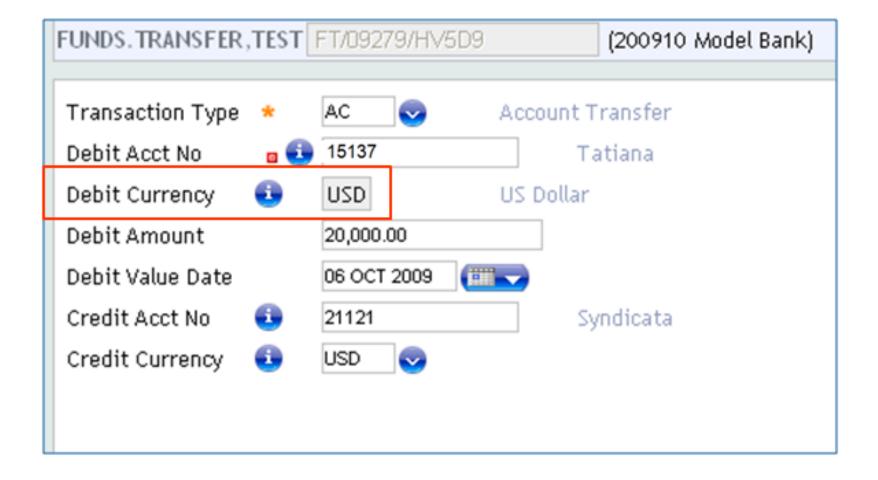






```
*Version routine to populate the debit currency
    SUBROUTINE VERSION. ROUTINE
    $INSERT I COMMON
    $INSERT I EQUATE
    $INSERT I F.ACCOUNT
    $INSERT I F.FUNDS.TRANSFER
        Y.ACCT.NO = COMI ; Y.ACCT.CUR = ''
        FN.ACCOUNT = 'F.ACCOUNT'; F.ACCOUNT = ''
        CALL OPF (FN.ACCOUNT, F.ACCOUNT)
        CALL F.READ (FN.ACCOUNT, Y.ACCT.NO, R.ACCT.REC, F.ACCOUNT, ERR1)
        Y.ACCT.CUR = R.ACCT.REC<AC.CURRENCY>
        R.NEW(FT.DEBIT.CURRENCY) = Y.ACCT.CUR
        * BUILD.SCREEN needed only for CUI, ignored for BROWSER
        CALL BUILD. SCREEN
    RETURN
    END
```





Debugger Mode



```
0011 DEBUG
jBASE debugger->S
             Y.ACCT.NO = COMI ; Y.ACCT.CUR = ''
0012
jBASE debugger->V COMI.
COMMON variables
  COMI
                              : 15137
|iBASE debugger->S
0013
             FN.ACCOUNT = 'F.ACCOUNT' ; F.ACCOUNT = ''
jBASE debugger->S
0014
             CALL OFF (FN. ACCOUNT, F. ACCOUNT)
jBASE debugger->S
0015
             CALL F.READ (FN.ACCOUNT, Y.ACCT.NO, R.ACCT.REC, F.ACCOUNT, ERR1)
jBASE debugger->S
             Y.ACCT.CUR = R.ACCT.REC<AC.CURRENCY>
0016
iBASE debudder->S
0017
             R.NEW(FT.DEBIT.CURRENCY) = Y.ACCT.CUR
jBASE debugger->V Y.ACCT.CUR
  Y.ACCT.CUR
                              : USD
|jBASE debugger->S
0018
             CALL REBUILD.SCREEN
jBASE debugger->
```

Good to Know



- COMI To default values in field in version validation routines.
- Though COMI is generally used for field validations. It is better to use COMI during commit phase as well



- BANK XYZ has some accounts as pyramid accounts. Entries should not be passed for any value date other than "Today" for these accounts for all FT transactions
- Developments: One Local Reference Field needs to be created in the Account application to indicate whether the account is a pyramid or not. Create a local reference field called 'PYRAMID' that can contain any one of the following 2 values only 'Y' or 'N'
- Additional Information: When an account is created, the user needs to input a value into this field PYRAMID. When a FT transaction is input, check to see if the DEBIT ACCOUNT NUMBER is a PYRAMID type of account. If so, check if the value date entered is the current date, else default to the current date

Input Routine



- Invoked prior to committing a transaction in unauthorized stage
- Invoked after validation processing
- Can get invoked even when the record is put on hold
- Routines may be used for additional checking or override processing
- Attached in the field INPUT.ROUTINE in VERSION application

Input Routines – Specific Common Variables



- COMI not available for Input Routines
- Values to be extracted from R.NEW
- ETEXT defined in I_COMMON used to store the error message
- STORE.END.ERROR is a subroutine used to display the error message
- SYNTAX: ETEXT = <"ERROR MESSAGE">

 CALL STORE.END.ERROR
- AF, AV and AS are used to display the error message next to a field

Task – Input Routine



- When a user commits a record in a version of the FUNDS.TRANSFER application, a check has to be made to see whether the Debit Account Currency and the Credit Account Currency are the same. If not, an error message "DEBIT AND CREDIT CURRENCY NOT EQUAL" has to be displayed
- Fields that need to be displayed as part of the version are

TRANSACTION. TYPE

DEBIT.ACCT.NO

DEBIT.CURRENCY

DEBIT.AMOUNT

DEBIT.VALUE.DATE

CREDIT.ACCT.NO

CREDIT.CURRENCY

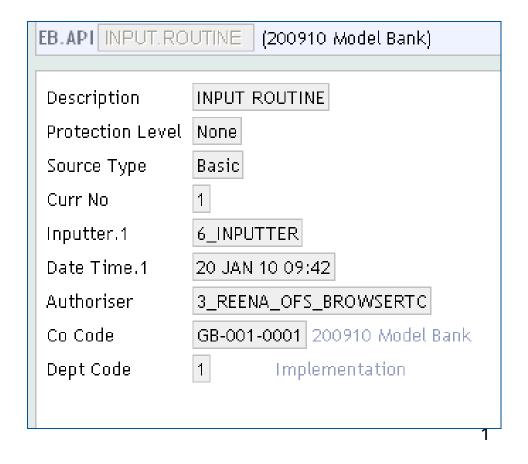


VERSION, FUNDS.TRANSFER, TES (200910 Mod Records Per Page Fields Per Line Language Code.1 **English** Field No.1 TRANSACTION.TYPE Field No.2 DEBIT.ACCT.NO Field No.3 DEBIT.CURRENCY Field No.4 DEBIT.AMOUNT Field No.5 DEBIT.VALUE.DATE Field No.6 CREDIT.ACCT.NO Field No.7 CREDIT.CURRENCY No Of Auth



```
*Version routine to checks if the debit and credit currency are
*equal.
SUBROUTINE INPUT.ROUTINE
    $INSERT I COMMON
    $INSERT I EQUATE
    $INSERT I F.FUNDS.TRANSFER
    Y.DEBIT.CUR = R.NEW(FT.DEBIT.CURRENCY)
    Y.CREDIT.CUR = R.NEW(FT.CREDIT.CURRENCY)
    IF Y.DEBIT.CUR NE Y.CREDIT.CUR THEN
        AF = FT.CREDIT.CURRENCY
        ETEXT = 'DEBIT AND CREDIT CURRENCY NOT EQUAL'
        CALL STORE END ERROR
    END
    RETURN
END
```

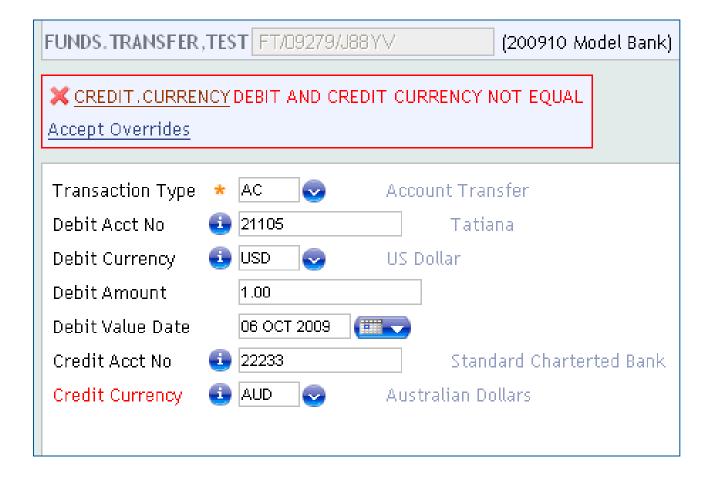




VERSION, FUNDS.TRANSFER,TES (200910 Model Bank)

Input Routine.1 INPUT.ROUTINE





The Banking Software Company



```
jBASE debugger->S
         Y.CREDIT.CUR = R.NEW(FT.CREDIT.CURRENCY)
0011
jBASE debugger->V Y.DEBIT.CUR
  Y.DEBIT.CUR
                             : USD
jBASE debugger->S
0012
         IF Y.DEBIT.CUR NE Y.CREDIT.CUR THEM
jBASE debugger->V Y.CREDIT.CUR
  Y.CREDIT.CUR
                             : AUD
jBASE debugger->S
0013
             AF = FT.CREDIT.CURRENCY
|iBASE debugger->S
0014
             ETEXT = 'DEBIT AND CREDIT CURRENCY NOT EQUAL'
jBASE debugger->V AF
COMMON variables
                             : 13
  AF
jBASE debugger->S
0015
             CALL STORE.END.ERROR
jBASE debugger->V ETEXT
COMMON variables
                             : DEBIT AND CREDIT CURRENCY NOT EQUAL
  ETEXT
```

Authorization Routine



- Invoked at the authorization stage
- Routine are used to update local files
- Invoked after validation process
- Attached in the field AUTH.ROUTINE in VERSION application
- Invoked after a write is made to the buffer
- Call to F.WRITE should precede JOURNAL.UPDATE



Once a record in a version of the FUNDS.TRANSFER application is authorized, the following details from the transaction need to be extracted and written on to a flat file named TEMENOS.TRAINING for interface purposes all delimited with a single space

DEBIT.ACCT.NO
CREDIT.ACCT.NO
DEBIT.VALUE.DATE
DEBIT.AMOUNT
DEBIT.CURRENCY

ID of the record in the flat file should be the ID of the FT record



VERSION, FUNDS.TRANSFER, TES (200910 Mod Records Per Page Fields Per Line Language Code.1 **English** Field No.1 TRANSACTION.TYPE DEBIT.ACCT.NO Field No.2 Field No.3 DEBIT.CURRENCY Field No.4 DEBIT.AMOUNT Field No.5 DEBIT.VALUE.DATE Field No.6 CREDIT.ACCT.NO Field No.7 CREDIT.CURRENCY No Of Auth

The Banking Software Company



Create a flat file under the HOME directory

```
jsh R10 ~ -->CREATE.FILE TEMENOS.TRAINING TYPE=UD
[ 417 ] File TEMENOS.TRAINING]D created , type = UD
[ 417 ] File TEMENOS.TRAINING created , type = UD
```

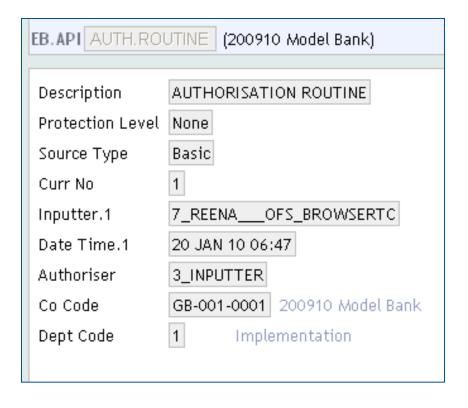


```
*AUTHORIZATION ROUTINE
SUBROUTINE AUTH.ROUTINE
    $INSERT I COMMON
    $INSERT I EQUATE
    $INSERT I F.FUNDS.TRANSFER
    Y.FILE.NAME = "TEMENOS.TRAINING"
    Y.FILE.ID = ID.NEW
    OPENSEQ Y.FILE.NAME, Y.FILE.ID TO Y.FILE.PTR ELSE
        CREATE Y.FILE.PTR ELSE
            CRT "Unable to create record in file"
        END
    END
```



```
Y.STRING.TO.BE.WRITTEN = R.NEW(FT.DEBIT.ACCT.NO):" ":
R.NEW (FT.DEBIT.CURRENCY)
Y.STRING.TO.BE.WRITTEN := " ": R.NEW(FT.CREDIT.ACCT.NO):" ":
R.NEW (FT.DEBIT.VALUE.DATE)
Y.STRING.TO.BE.WRITTEN := " ": R.NEW(FT.DEBIT.AMOUNT):" ":
R.NEW (FT. CREDIT. CURRENCY)
    WRITESEO Y.STRING.TO.BE.WRITTEN APPEND TO Y.FILE.PTR ELSE
        CRT "Unable to write to the file"
    END
    RETURN
 END
```

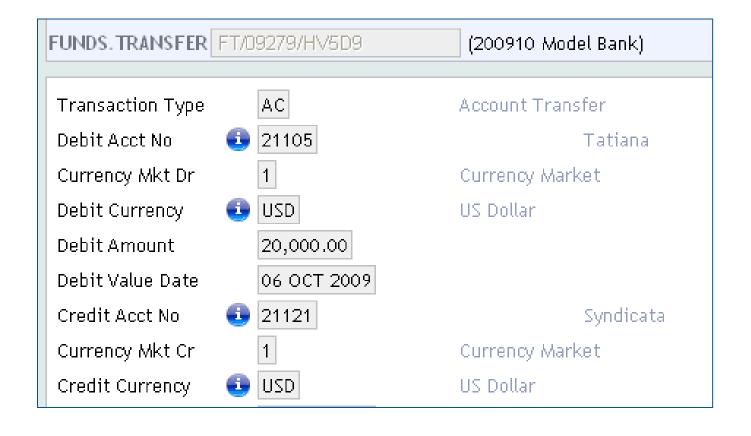




VERSION, FUNDS.TRANSFER,TE3 (200910 Model Bank)

Auth Routine.1 AUTH.ROUTINE







LIST TEMENOS.TRAINING

TEMENOS. TRAINING

FT09279HV5D9

1 Records Listed



- Once a record in a version of the CUSTOMER application is authorised, the following details from the transaction need to be extracted and written on to a flat file named TEMENOS.XX.CUS.INTERFACE for interface purposes.
 - XX Is your name
 - ID of the record in the flat file needs to be the Customer ID
 - Following field's values need to be stored in the flat file all delimited with a single space
 - Mnemonic
 - Sector
 - Nationality
 - Language
- The above operation needs to be done only if the record being authorised is a new customer record

Number of authoriser in Version



- Can control the number of authorisers
- Allowed values 0, 1, or 2
- AUTH.NO common variable

Common variables for VERSION



- PGM.VERSION holds the version name
- R.VERSION Version record



QUIZ



In this learning unit/course, you learnt about the basic version routines in T24

You will now be able to:

- Explain the need for version routines
- Create different version routines

Thank You

TEMENOS EDUCATION CENTRE

NOTICE

These training materials are the copyrighted work of Temenos Headquarters SA and other companies in the TEMENOS group of companies (The Copyright Owner). The training materials contain protected logos, graphics and images. Use of the training materials is restricted solely for use by licensed end users, partners and employees. Any un-licensed reproduction by any means, redistribution, editing, transformation, publishing, distribution, or public demonstration of the training materials whether for commercial or personal gain is expressly prohibited by law, and may result in severe civil and criminal penalties. Violators will be prosecuted to the maximum extent possible. Such training materials shall not be represented, extracted into or included in part, or in whole, as part of any other training documentation without the express permission of the Copyright Owner, which must given in writing by an authorised agent of the Copyright Owner to be valid. Where such permission is given a clear and prominent notice must be displayed on any and all documentation accrediting the Copyright Owner with having copyright over the materials. End-user licenses will in no event contain permissions extending the use of these training materials to third parties for commercial training purposes.

Without limiting the foregoing, copying or reproduction of the training materials in part or in whole to any other sever or location for further reproduction or redistribution is expressly prohibited, unless such reproduction is expressly licensed by the Copyright Owner.

Copyright © 2010 Temenos Headquarters SA













