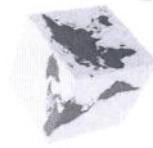


Deal Slip Routines

TEMENOS EDUCATION CENTRE

Warning: This document, is protected by copyright law and international treaties. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TEMENOS HEADQUARTERS SA. Unauthorized reproduction or distribution of this presentation or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under applicable law." Information in this document is subject to change without notice



TEMENOS

The Banking Software Company



TEMENOS

The Banking Software Company

At the end of the session you will have

- Adequate knowledge to understand
 - Need for a deal slip routine
 - Steps to attach a deal slip routine to DEAL.SLIP.FORMAT application
- Adequate knowledge to create
 - Deal slip routines

- Deal slip routines are similar to version validation routines
- They are attached to specific fields in the deal slip
- Get invoked after the data is picked up for the specific field but before the field's data can be displayed

The application CHEQUE.ISSUE is used to issue cheque books. When cheque books are issued to customers, the account number and the account title need to be printed on the cheque leafs in the following format. Pre printed stationery with the Banks name is available. Only the Account number and the Account title needs to be printed.

ABN Amro Bank		05/05/05
Pay	Kripesh Krishnadas Five Hundred Thousand Dollars	a sum of only

Account No : 70087	Account Name : ANDRE LOUSTAU	

The Banking Software Company

CHEQUE.TYPE (BIN 200712 002)	
OB Description	Current Account type
Min Holding	5
Max Holding	500
Default Issue No	100
Category 1	1001
Allow Fcy Acct	YES

The Banking Software Company

- Open the account file
- Read the record pertaining to the specific account number (this account number will have to be supplied as an input parameter to the deal slip routine)
- From the record, extract the account name and pass it back to the DEAL.SLIP.FORMAT application through the subroutine

All deal slip routines take in one parameter

- Where do we attach the subroutine ?
- Application : DEAL.SLIP.FORMAT
- Field : FORMAT
- Name of the subroutine needs to be prefixed with an @

- Write, compile and catalog

```
SUBROUTINE TRG.DS.CHQ.ISSUE.RTN(ACCOUNT.NAME)

* This Subroutine is written to get the ACCOUNT.NO as the incoming
* parameter and using that reads the ACCOUNT File and extracts the
* Account name for the specific number. The extracted Account name
* is passed back to the deal slip as an output parameter.
* This Routine will be attached in the DEAL.SLIP.FORMAT
* record for CHEQUE.ISSUE.

$INSERT I_COMMON
$INSERT I_EQUATE
$INSERT I_F.ACCOUNT

AC.ERR = ''
FN.AC = 'F.ACCOUNT'
F.AC = ''
CALL OPF(FN.AC,F.AC)
Y.ACCOUNT.NUMBER = ACCOUNT.NAME
CALL F.READ(FN.AC,Y.ACCOUNT.NUMBER,R.AC.REC,F.AC,AC.ERR)
ACCOUNT.NAME = R.AC.REC<AC.ACCOUNT.TITLE.1>
RETURN
END
```

- Make an entry in the PGM.FILE with field TYPE set to 'S'

PGM.FILE, TRG.05.CHEQUEISSN.RTM

TYPE	S
OB Screen Title	Cheque Issue Debit/Routine
Product	EB
Curr No	1
Inputter.1	E_INPUTTER_OFS_GCS
Date/Time.1	06 MAR 09 09 41
Authoriser	E_INPUTTER_OFS_GCS
Co Code	BB-001-0001 RBT INDIA MODEL BANK
Dept Code	1

The Banking Software Company

Create the record in DEAL SLIP FORMAT

TEMENOS

QB Description	Cheque Issue Deal Slip
Report Control Id	CHEQUEISSUE RC
File.1	CHEQUE ISSUE
id.1	ID
Field.1	ABN Amro Bank
Column.1	2E.1
Format.1.1	13L
Field.2	SYSTEM DATE
Column.2	60.1
Format.2.1	00E
Field.3	Pay Kitlesh Krishnadas
Column.3	6.3
Format.3.1	27L
Field.4	a sum of
Column.4	60.3
Format.4.1	11L
Field.5	Five Hundred Thousand Dollars
Column.5	17.4
Format.5.1	21L
Field.6	only
Column.6	60.4
Format.6.1	6L

The Banking Software Company

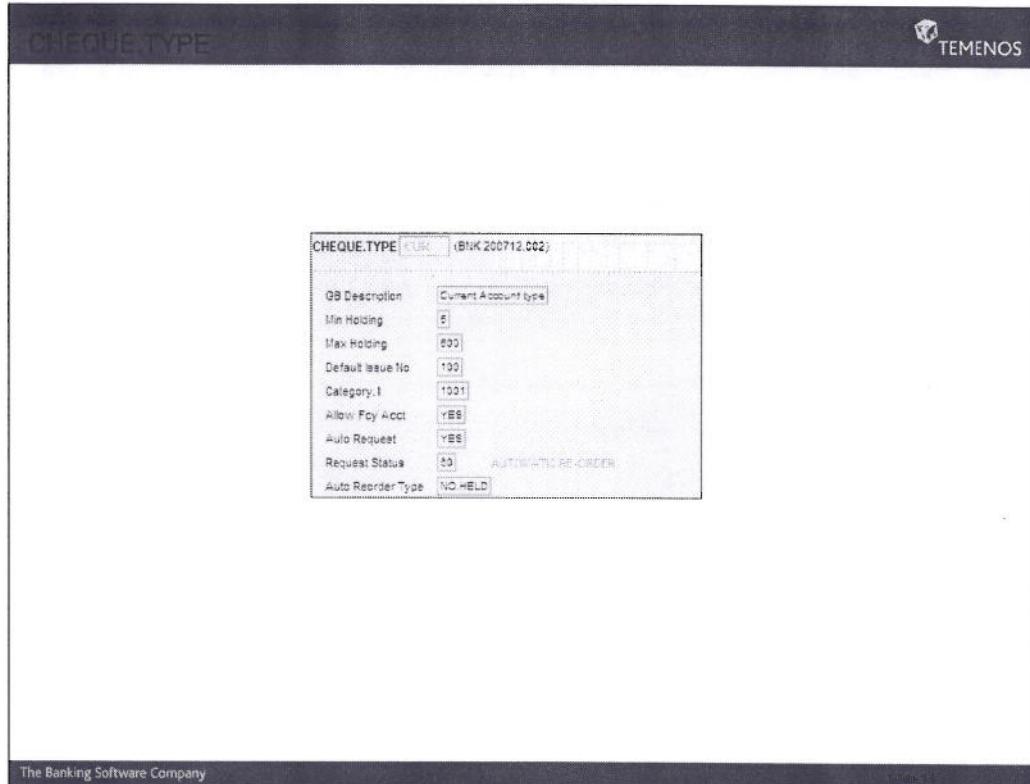
Attach the routine to the DEAL.SLIP.FORMAT application

TEMENOS

DEAL.SLIP.FORMAT [CHEQUEISSUEITS] (BRK 200712.002)

Column.6	80.4
Format.6.1	EL
Field.7	
Column.7	1.7
Format.7.1	3EL
Field.8	
Column.8	3E.7
Format.8.1	3EL
Field.9	
Column.9	70.7
Format.9.1	2EL
Field.10	Account No.
Column.10	18
Format.10.1	12
Field.11	ACCOUNT NO.
Column.11	15.5
Format.11.1	15
Field.12	Account Name.
Column.12	30.5
Format.12.1	TEL
Field.13	ACCOUNT NO.
Column.13	45.5
Format.13.1	STR32520 ISSUE PTS

The Banking Software Company



CHEQUE ISSUE

TEMENOS

CHEQUE ISSUE	CHECK YOUR BANKBOOK	CURRENT ACCOUNTS, ANDRE (BNK 200712.002)
Cheque Status	92	ISSUED
Issue Date	25 NOV 2001	
Number Issued	1	
Currency	USD	
Chq No Start	1	
Notes	TESTING CHEQUE SLIP	

The Banking Software Company

ABN Amro Bank		05/03/08
Pay	Kripesh Krishnadas Five Hundred Thousand Dollars	a sum of only
----- Account No : 70057		Account Name : ANDRE LOUSTAU

The Banking Software Company

- Deal slip routines are attached to the DEAL.SLIP.FORMAT application
- They take in one parameter
- They need to have an entry in the PGM.FILE application with the field TYPE set to S

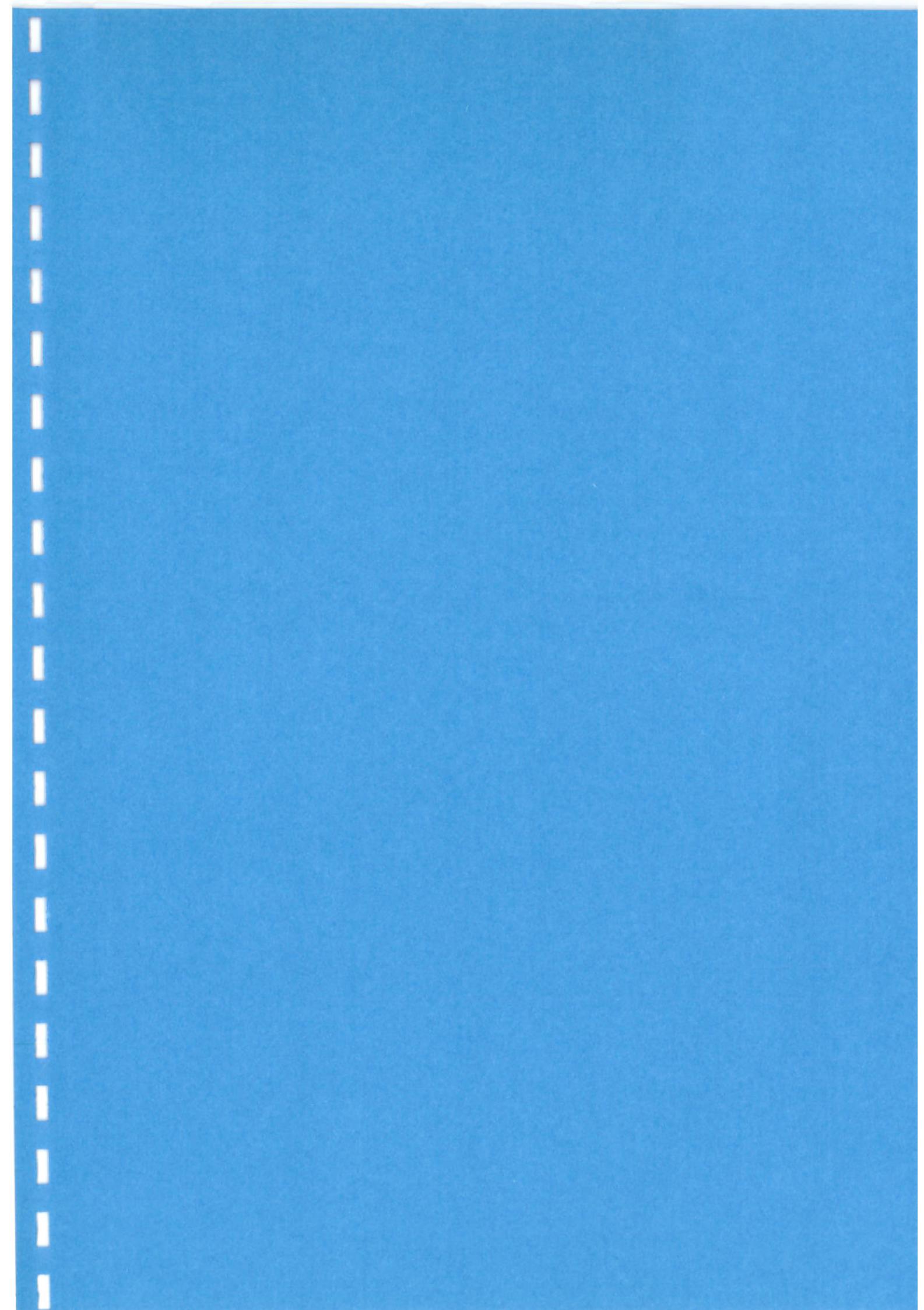
Thank You

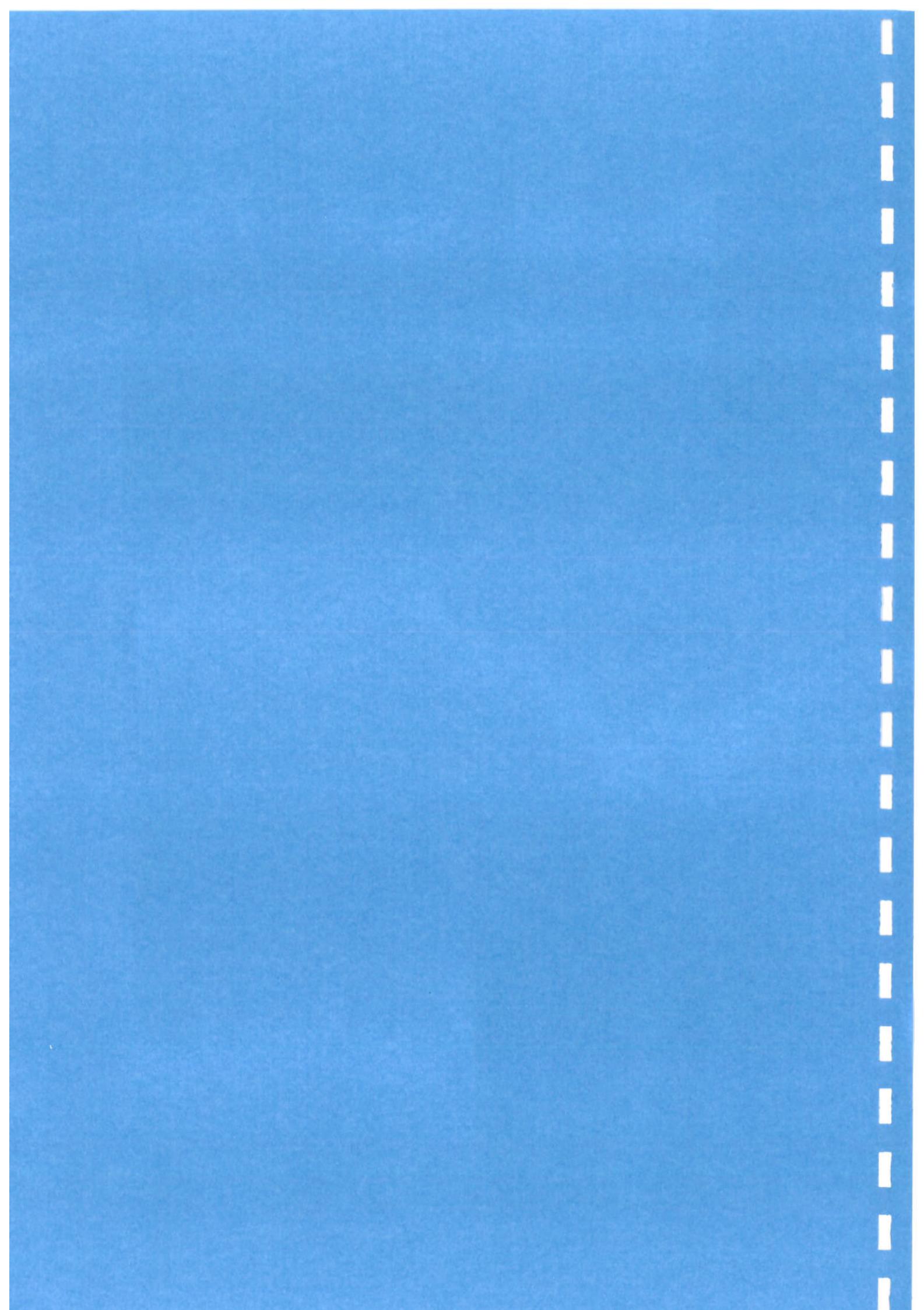
TEMENOS EDUCATION CENTRE

Warning: This document, is protected by copyright law and international treaties. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TEMENOS HEADQUARTERS SA. Unauthorized reproduction or distribution of this presentation or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under applicable law.* Information in this document is subject to change without notice.



TEMENOS
The Banking Software Company





Open Financial Service

TEMENOS EDUCATION CENTRE

Warning: This document, is protected by copyright law and international treaties. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TEMENOS HEADQUARTERS SA. Unauthorized reproduction or distribution of this presentation or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under applicable law." Information in this document is subject to change without notice



TEMENOS

The Banking Software Company



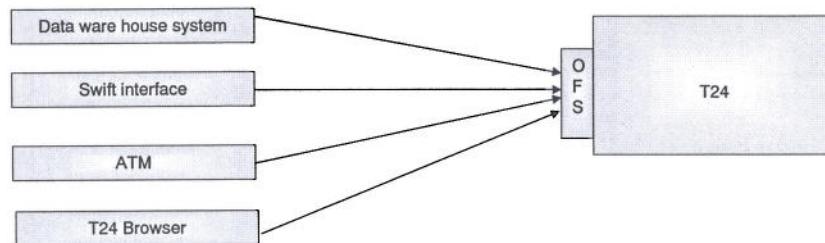
TEMENOS

The Banking Software Company

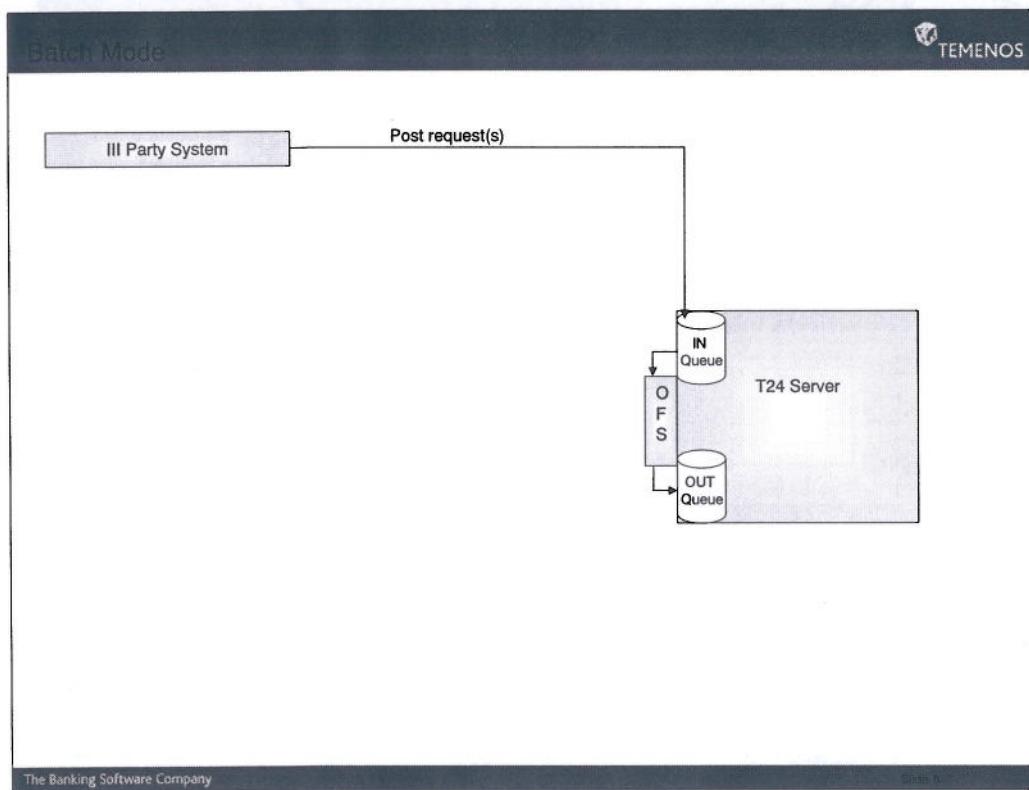
At the end of the session you will have

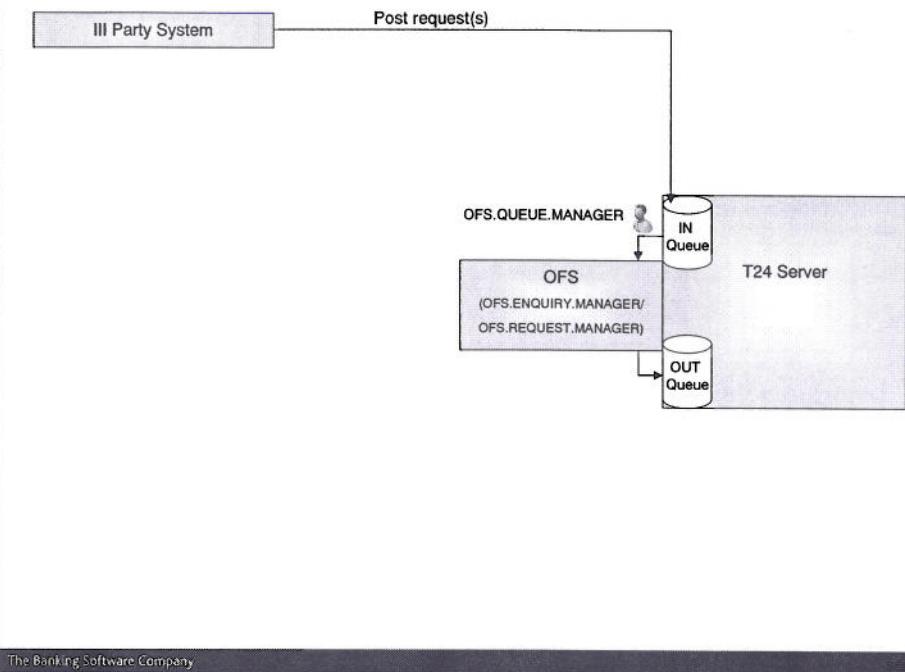
- Adequate knowledge to understand
 - The need for OFS
 - OFS modes
 - OFS message syntax
- Adequate knowledge to create
 - OFS requests
- Adequate knowledge to work on
 - OFS module in sufficient detail

- Open Financial Service (OFS)
 - Module in T24
 - Standard interface to and from T24
 - Works based on a request – response based system
- Enables III party systems to post requests and obtain responses



- Batch processing
 - BATCH mode
- Inter application processing (OFS.POST.MESSAGE)
 - GLOBUS mode
- Online processing
 - TELNET mode
 - SESSION mode

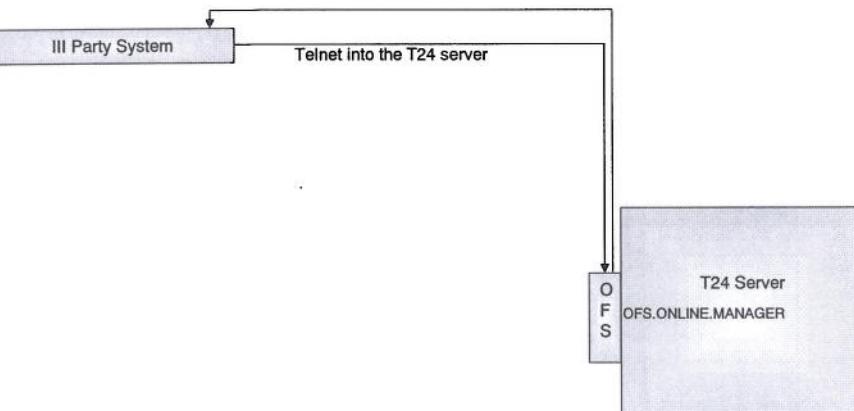




The Banking Software Company

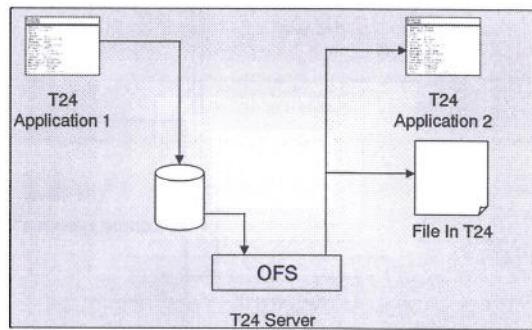
Telnet Mode (Almost Obsolete)

 TEMENOS



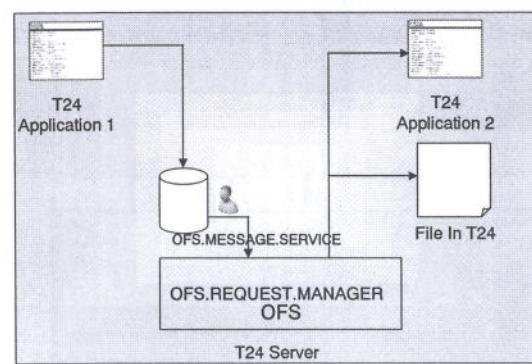
The Banking Software Company

Inter Application Calls (OFS POST MESSAGE)

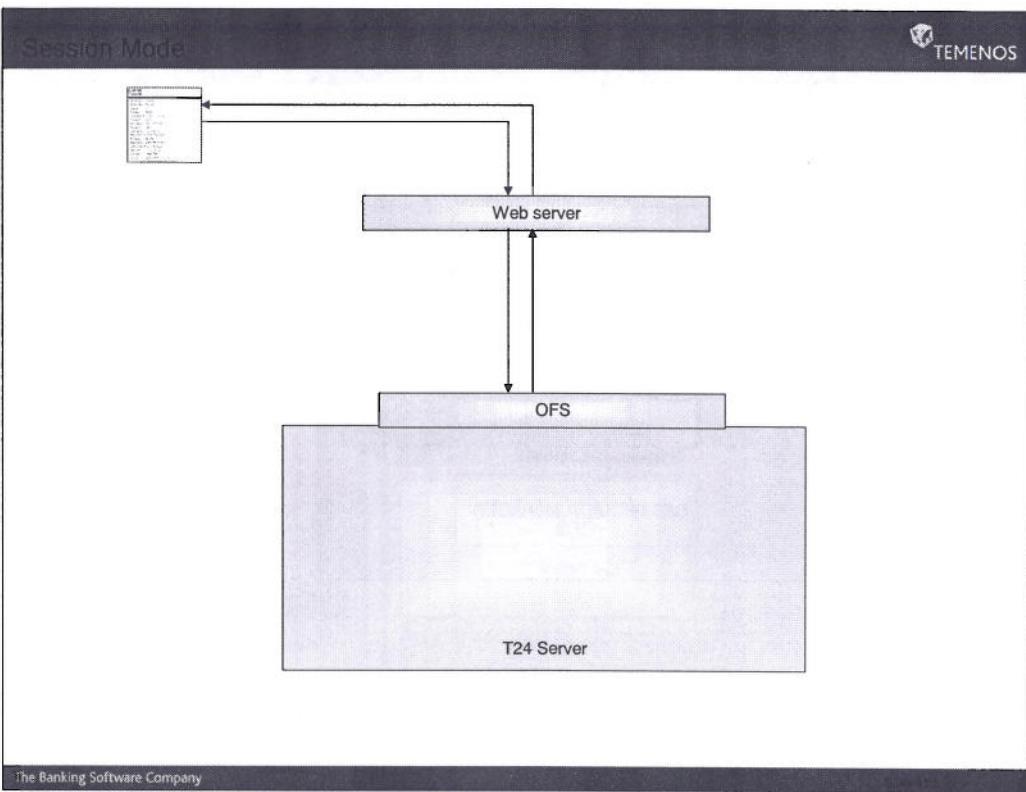


The Banking Software Company

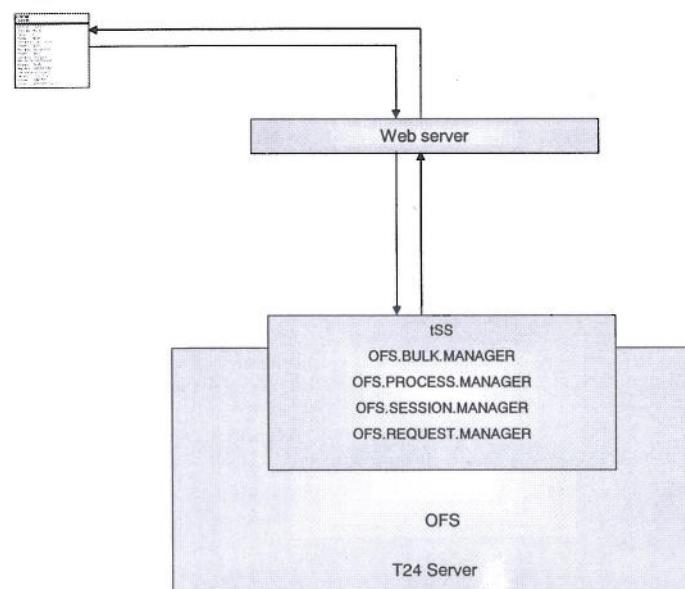
Inter Application Calls (OFS POST MESSAGE) – An Insight



The Banking Software Company



Session Mode – An Insight

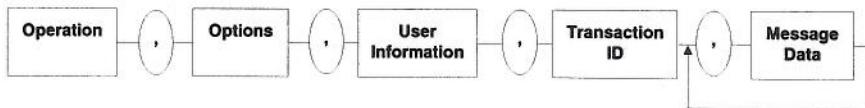


The Banking Software Company

- Transaction Request
- Enquiry Request

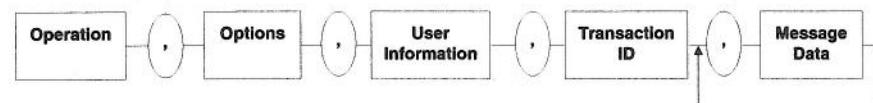
- Every transaction request that is used to insert or update a record in T24 should contain values for all mandatory fields.

```
ACCOUNT,SAMPLE/I/PROCESS/10000/2,  
INPUTT/123456,34567,CUSTOMER=100724,CATEGORY=1001,CURRENCY=USD
```



- Operation
 - Contains the name of APPLICATION.
 - Example: ACCOUNT.
- Options
 - Comprises of VERSION-NAME/FUNCTION/VALIDATE or PROCESS/GTS.CONTROL value/NO.OF.AUTHORISERS.
 - Example : TRG/I/VALIDATE//2
- User Information
 - This comprises of SignOnName/Password/Company code
 - E.g INPUTT/123456/GB0010001

Message Syntax



- **Transaction ID**
 - Contains transaction id of the record used in the transaction. May also contain an optional message id.
- **Message Data**
 - Contains the data required to create or update the transaction.
 - Eg: CUSTOMER=100724,CATEGORY=1001,CURRENCY=USD

The Banking Software Company

Write down the OFS message to create an account record in T24

Write down the OFS message to create a record in SECTOR application in T24

Applications To Set Up



- OFS.SOURCE
- EB.PHANTOM

The Banking Software Company

© 2008 Temenos

- OFS.SOURCE contains the parameters for setting up a OFS connection into T24.
- Usually different OFS source records are used for each connection

OF SOURCE

OF SOURCE: TIGEEXAMPLE

Description	BATCH FOR BATCH
Source Type	BATCH
Log File Dir	TRGLOG
Log Detail Level	FULL
Mail Mag Dets	N
Def Prefix	TRG
In Queue Dir	TRGIN
Out Queue Dir	TRGOUT
Syntax Type	OFIS
Generic User	SHERIFFA1
Same Authoriser	YES
Curr No	1
Inputter.1	S_SHERIFFA1_OFIS_BROWSETC
Date Time.1	23 NOV 07 09:57
Authoriser	S_SHERIFFA1_OFIS_BROWSETC
Ref No	2841114311

The Banking Software Company

0.000.000

Key Fields To Understand Now

Id

Source Type

In Queue Dir

Out Queue Dir

Syntax Type

- EB.PHANTOM controls all phantom and interactive jobs in T24.
- The control parameters for phantoms such as the name of the program to execute, sleep time etc are specified here
- We need to make OFS.REQUEST.MANAGER run as a phantom in order for OFS to pick up requests from the IN queue and process it.

EB.PHANTOM

TEMENOS

Key Fields To Understand Now

Id

Status

Phant Stop Req

Run Mode

OFS.SOURCE

Sleep secs

GTS user ID

The Banking Software Company

- After creating records in OFS.SOURCE and EB.PHANTOM the next step is to create the request

```
ACCOUNT, SAMPLE/I/PROCESS,  
INPUTT/654321,,CUSTOMER=100724,CATEGORY=1001,  
CURRENCY=USD
```

Note:

- All the mandatory fields of the application have to be filled.
- The id will be automatically generated by the system.
- We use a version called SAMPLE which has number of authorizers set to 1

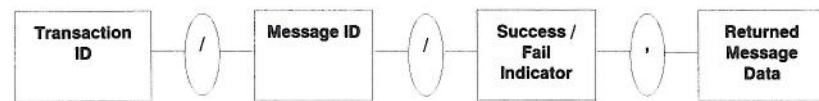
- Place the file that contains the request in the designated IN queue directory
- Verify the record in EB.PHANTOM.
- Once processed, the request will be moved from TRGIN to TRGOUT.

- An account record is created in the Account file with record status INAU. The transaction response looks like this

```
35114/TRG0723600007/1,CUSTOMER=100724:1:1,CATEGORY=1001:1:1,ACCO  
UNT.TITLE.1=AAA SHIPPING COMPANY OF PANAMA:1:1,SHORT.TITLE=AAA  
SHIPPING COMPANY OF PANAMA:1:1, POSITION.TYPE=TR:1:1,  
CURRENCY=USD:1:1, CURRENCY.MARKET=1:1:1, ACCOUNT.OFFICER=27:1:1  
, CONDITION.GROUP=2:1:1,PASSBOOK=NO:1:1,OPEN.CATEGORY=1001:1:1,CH  
ARGE.CCY=USD:1:1,INTEREST.CCY=USD:1:1,ALT.ACCT.TYPE=LEGACY:1:1,A  
LLOW.NETTING=NO:1:1,SINGLE.LIMIT=Y:1:1,  
RECORD.STATUS=INAU:1:1,CURR.NO=1:1:1,INPUTTER=8_BUILDUSER90____OF  
S_TRG.EXAMPLE:1:1,DATE.TIME=0711271133:1:1,CO.CODE=GB0010001:1:1  
,DEPT.CODE=1:1:1
```

The Banking Software Company

OFS Message Syntax – Transaction Response Format



The Banking Software Company

- **TRANSACTION ID**
 - The transaction ID contains either the value supplied for the transaction in the request or the value that is automatically generated by the TEMENOS T24 application (when no value is supplied in the request).
- **MESSAGE ID**
 - The Message ID contains the value of Message ID if supplied in the request.

■ SUCCESS / FAIL INDICATOR

■ Indicates the status of the transaction request processed. OFS returns one of the following values:

- 1 Successful transaction.
- -1 Errors encountered during processing.
- -2 Override condition (s) encountered during processing.
- -3 T24 server is offline

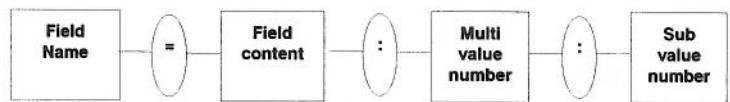
■ RETURNED MESSAGE DATA

■ A successfully processed message will contain all the fields populated in the transaction.

■ The format is the same as that of the request message data.

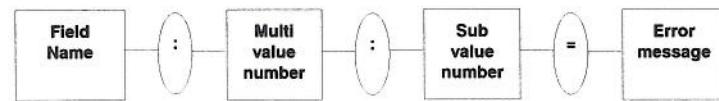
■ However for a unsuccessful transaction, the returned message data has a slightly different format.

Transaction response – Returned Message Data



The Banking Software Company

Transaction Response – Returned Message Data With Errors



The Banking Software Company

- How do we authorise?
 - To authorise the transaction, use the following message syntax.
- ACCOUNT , /A/PROCESS , INPUTT/ 654321 , 35114
- Note - We could have used a zero authoriser version to input and authorise the transaction in one shot.
 - Did the record get authorised when you use the same EB.PHANTOM that was used to input? No? Why not?

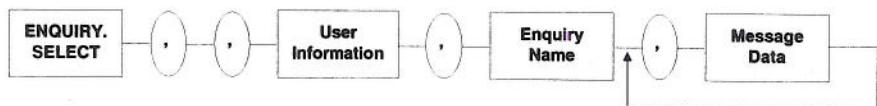
- When you input a transaction using OFS (In any mode), based on the no of authorizers set in the version, the record will either get to INAU or live status.
- In this case the record is in INAU status and the record has been input using the BATCH mode of T24.
- Now, if you send another message via OFS to authorize this record (using the same BATCH), you will get an error message "**EB.RTN.SAME.NAME.AUTHORISER/INPUTTER**"

- To overcome this, set the field SAME.AUTHORISER in OFS.SOURCE to YES.
- Same authorizer field is quite useful as the same EB.PHANTOM record can be used for input as well as authorization.

Discuss : What other solutions can we use?

- Create a Transaction type request to INPUT a SECTOR record. Use a Single Authorizer VERSION.
- Create a Transaction type request to Authorize the SECTOR record created in the previous step.
- Create a Transaction type request to Input a CUSTOMER. Create and use a Zero Authorizer VERSION.
- Create a Transaction type request to Input a new ACCOUNT for the CUSTOMER created above. Do not use a VERSION record.

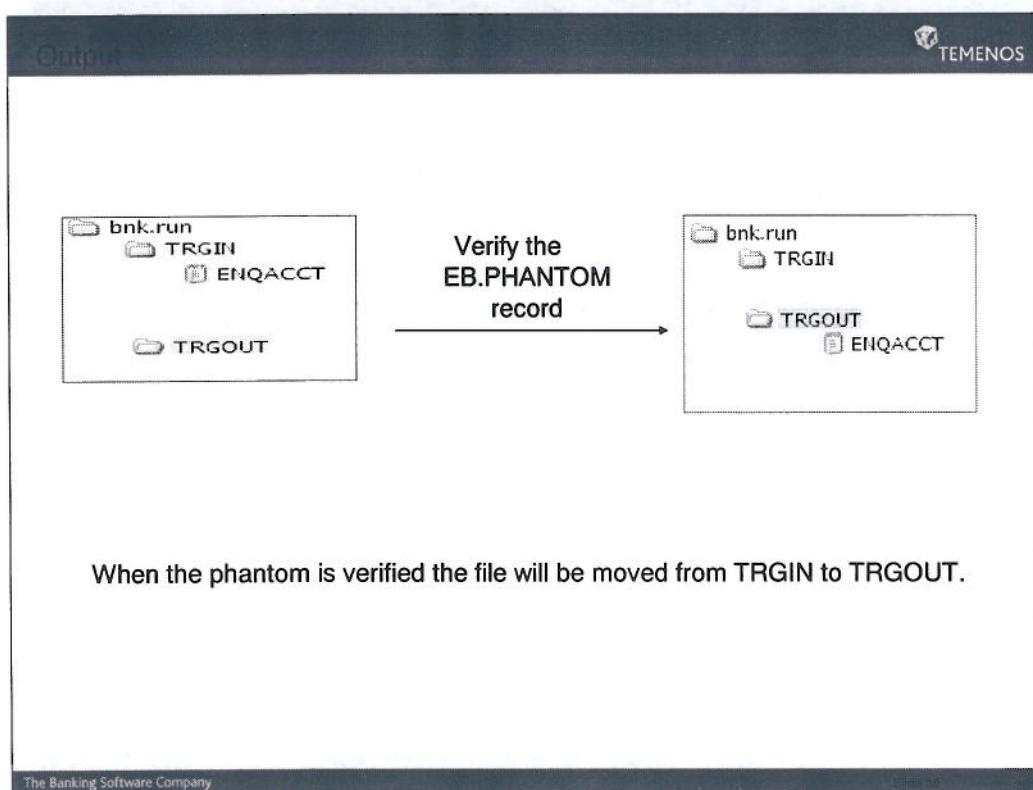
OFS Message Syntax – Enquiry request



The Banking Software Company

- ENQUIRY.SELECT
 - The first portion of an enquiry type request must always be ENQUIRY.SELECT. This is the name of the routine that is used to run queries and return the data.
- USER INFORMATION
 - The user information portion of the message structure is same as that in the transaction type request.
- ENQUIRY.NAME
 - Name of the T24 Enquiry that will be run.
 - The Enquiry name supplied here must be a valid TEMENOS T24 enquiry (i.e. must be found in the ENQUIRY application of TEMENOS T24).
- MESSAGE DATA
 - The message data portion of the enquiry message structure contains the selection criteria passed to the enquiry.
 - The message data portion of the message can be repeated for each selection criteria separated by a comma (,).
 - This is optional depending on the Enquiry

- We will illustrate Enquiry requests by executing the default ACCOUNT enquiry.
- ENQUIRY.SELECT,,INPUTT/654321,%ACCOUNT**
- Place the Enquiry Request in a file named ENQACCT and place the file under the designated IN Queue
 - Verify the record in EB.PHANTOM
 - The enquiry %ACCOUNT will get executed and the enquiry response can be viewed in file ENQACCT under the OUT Queue



- Create a Enquiry type request to find the list of INDUSTRY records found in T24 environment.
- Create an Enquiry type request to find the current day's balance summary of an ACCOUNT (say for example 29987). (TIP: Use the enquiry ACCT.BAL.TODAY)

- We will look at the following message features next
 - No of authorizers
 - Replace option
 - Unique Message Reference
 - Logging using ORD

- We can use a //0 after PROCESS to indicate no of authorisers.
- This overrides the no of authorisers specified in the version (if a version is used)

```
CUSTOMER,/I/PROCESS//0,AUTHOR/123456,,MNEMONIC=AIRFRG,SHO
RT.NAME=AIRBOURNE FREIGHT,NAME.1=AIRBOURNE
FREIGHT,STREET=3101 WESTERN
AVE,TOWN,COUNTRY=SEATTLE,RELATION.CODE:1=1,REL.CUSTOMER:1
=100424,RELATION.CODE:2=2,REL.CUSTOMER:2=100724,SECTOR=20
01,ACCOUNT.OFFICER=1,INDUSTRY=1000,TARGET=999,NATIONALITY
=IN,CUSTOMER.STATUS=1,RESIDENCE=IN,LANGUAGE=1
```

```
100966/IM0723600001/1,MNEMONIC=AIRFRG:1:1,SHORT.NAME=AIRB  
OURNE FREIGHT:1:1,NAME.1=AIRBOURNE  
FREIGHT:1:1,STREET=3101 WESTERN  
AVE:1:1,TOWN.COUNTRY=SEATTLE:1:1,RELATION.CODE=1:1:1,RELA  
TION.CODE=2:2:1,REL.CUSTOMER=100424:1:1,REL.CUSTOMER=1007  
24:2:1,REVERS.REL.CODE=10:1:1,REVERS.REL.CODE=12:2:1,SECT  
OR=2001:1:1,ACCOUNT.OFFICER=1:1:1,INDUSTRY=1000:1:1,TARGE  
T=999:1:1,NATIONALITY=IN:1:1,CUSTOMER.STATUS=1:1:1,RESIDE  
NCE=IN:1:1,LANGUAGE=1:1:1,COMPANY.BOOK=GB0010001:1:1,CLS.  
CPARTY=NO:1:1,OVERRIDE=INTRO/CUS*100 FROM 100966 NOT  
RECEIVED:1:1,CURR.NO=1:1:1,INPUTTER=15_AUTHORISER____OFS_T  
AABS:1:1,DATE.TIME=712131510:1:1,AUTHORISER=15_AUTHORISER  
_OFS_TAABS:1:1,CO.CODE=GB0010001:1:1,DEPT.CODE=1:1:1
```

- From R7 there is a special option, that will allow records to be cleared and re-input.
- This option when used, clears existing multi-values also
- Lets illustrate by using the CUSTOMER record created previously.
- This has 2 multivalue's for the fields Relation.code, Rel.customer.

```
Relation.code.1=1  
Rel.customer.1=100424
```

```
Relation.code.2=2  
Rel.customer.2=100724
```

Transaction message without replace option



```
CUSTOMER,/I/PROCESS//0,AUTHOR/123456,100966,REL.CUSTOMER:1=100172
```

```
100966/IM0723600003/1,MNEMONIC:1:1=AIRFRG,SHORT.NAME:1:1=AIRBOURNE FREIGHT,NAME.1:1:1=AIRBOURNE FREIGHT,STREET:1:1=3101 WESTERN AVE,TOWN.COUNTRY:1:1=SEATTLE,RELATION.CODE:1:1=1,RELATION.CODE:2:1=2,REL.CUSTOMER:1:1=100172,REL.CUSTOMER:2:1=100724,REVERS.REL.CODE:1:1=10,REVERS.REL.CODE:2:1=12,SECTOR:1:1=2001,ACCOUNT.OFFICER:1:1=1,INDUSTRY:1:1=1000,TARGET:1:1=999,NATIONALITY:1:1=IN,CUSTOMER.STATUS:1:1=1,RESIDENCE:1:1=IN,LANGUAGE:1:1=1,COMPANY.BOOK:1:1=GB0010001,CLS.CPARTY:1:1=NO,OVERRIDE:1:1=INTRO/CUS*100 FROM 100966 NOT RECEIVED,CURR.NO:1:1=2,INPUTTER:1:1=15_AUTHORISER____OFS_TAABS,DATE.TIME:1:1=0712131515,AUTHORISER:1:1=15_AUTHORISER_OFS_TAABS,CO.CODE:1:1=GB0010001,DEPT.CODE:1:1=1
```

Note that only the specific multi-value is changed

The Banking Software Company

```
CUSTOMER,/I/PROCESS//0,AUTHOR/123456///1,100966,MNEMONIC  
=AIRFRG,SHORT.NAME=AIRBOURNE FREIGHT,NAME.1=AIRBOURNE  
FREIGHT,STREET=3101 WESTERN  
AVE,TOWN,COUNTRY=SEATTLE,RELATION.CODE:1=1,REL.CUSTOMER:1  
=100300,SECTOR=2001,ACCOUNT.OFFICER=1,INDUSTRY=1000,TARGE  
T=999,NATIONALITY=IN,CUSTOMER.STATUS=1,RESIDENCE=IN,LANGU  
AGE=1
```

- **USER INFORMATION**
 - Username/password/companycode///1
 - 1 denotes replace option.

Transaction Response



```
100966/IM0723600005/1,MNEMONIC=AIRFRG:1:1,SHORT.NAME=AIRB  
OURNE FREIGHT:1:1,NAME.1=AIRBOURNE  
FREIGHT:1:1,STREET=3101 WESTERN  
AVE:1:1,TOWN.COUNTRY=SEATTLE:1:1,RELATION.CODE=1:1:1,REL.  
CUSTOMER=100300:1:1,REVERS.REL.CODE=10:1:1,SECTOR=2001:1:  
1,ACCOUNT.OFFICER=1:1:1,INDUSTRY=1000:1:1,TARGET=999:1:1,  
NATIONALITY=IN:1:1,CUSTOMER.STATUS=1:1:1,RESIDENCE=IN:1:1  
,LANGUAGE=1:1:1,COMPANY.BOOK=GB0010001:1:1,CLS.CPARTY=NO:  
1:1,OVERRIDE=INTRO/CUS*100 FROM 100966 NOT  
RECEIVED:1:1,CURR.NO=3:1:1,INPUTTER=15_AUTHORISER____OFS_T  
AABS:1:1,DATE.TIME=0712131521:1:1,AUTHORISER=15_AUTHORISE  
R_OFS_TAABS:1:1,CO.CODE=GB0010001:1:1,DEPT.CODE=1:1:1
```

Note : The existing multi-valued REL.CUSTOMERs are replaced totally

```
CUSTOMER,/I/PROCESS//0,AUTHOR/123456///1,100966,  
REL.CUSTOMER:1=100172
```

```
100966/IM072360004/-1/NO,REL.CUSTOMER:1:1=MISSING  
CUSTOMER - RECORD,MNEMONIC:1:1=INPUT  
MISSING,SHORT.NAME:1:1=INPUT  
MISSING,REL.CUSTOMER:1:1=MISSING CUSTOMER -  
RECORD,REL.CUSTOMER:1:1=MISSING CUSTOMER -  
RECORD,SECTOR:1:1=INPUT MISSING,LANGUAGE:1:1=INPUT  
MISSING,REL.CUSTOMER:1:1=RELATION.CODE  
MISSING,STREET:1:1=INPUT MANDATORY FOR GIVEN  
SECTOR,NAME.1:1:1=Input NAME.1 OR GIVEN.NAMES OR  
FAMILY.NAME,GIVEN.NAMES:1:1=Input NAME.1 OR GIVEN.NAMES  
OR FAMILY.NAME,FAMILY.NAME:1:1=Input NAME.1 OR  
GIVEN.NAMES OR FAMILY.NAME
```

- OFS Unique Message Reference is to be supplied for all OFS Transaction requests in a non Browser environment.
- This is a Temenos recommendation
- OFS Unique Message Reference – This unique number can be sent as part of the OFS message.
- How?

```
ACCOUNT=SAMPLE/I/PROCESS,INPUTT/654321,34343/55555,  
CUSTOMER=100424,CATEGORY=1001,CURRENCY=USD.
```

- The highlighted section of the above OFS Message is the OFS Unique Reference Number

- OFS.REQUEST.DETAILS sometimes referred to as ORD is used by OFS to maintain an audit of OFS requests.
- To enable the OFS audit in to this file, the field MAINT.MSG.DETS in OFS.SOURCE must be set to be "Y".
- The audit information is recorded per message.
- OFS.REQUEST.DETAIL is a live file and hence cannot be modified by the user .
- The field DET PREFIX in the OFS.SOURCE application controls the prefix for the records created in this application. This is an optional setting
- The OFS module automatically updates this file.

Fields in OFS.REQUEST.DETAIL

OFS.REQUEST.DETAIL [5F85200001259992.00] (R8 MODEL BANK)	
Application	SEC.OPEN.ORDER
Version	OFS
Function	R
Trans Reference	OPODSC0733900050
User Name	SUSER1
Company	680010001
Date Time Recd	16:39:29 03 APR 2008
Date Time Proc	16:39:29 03 APR 2008
Status	ERROR
Msg In	SEC.OPEN.ORDER,OFS/R/PROCESS,SUSER1*****/680010001,OPODSC0733900050,
Msg Out	OPODSC0733900050/STP08000001259992.00/1/NO,@ID:1=RECORD MISSING

ID:
What is the log record id.
Format : DET.PREFIX from OFS.SOURCE : Julian Date : Unique time

APPLICATION:
What application gets updated

FUNCTION:
What function is used

TRANS REFERENCE:
Which transaction (record) is affected

The Banking Software Company

- Switch on the OFS Request Detail in your OFS.SOURCE
-
- Write a OFS transaction request to create an account with two OTHER.OFFICER fields. Use a unique message reference also.
- Use the replace option in a second OFS message to replace these Other Officer fields with a single Other Officer field.
 - Do not use a unique message reference for this message
 - Note down what is returned in place of the message reference?
- View the log details

- Setting up and testing online and session connection
- We will discuss the OFS.SOURCE settings and then discuss sending a message

OFS SOURCE entries for SESSION mode

 TEMENOS

OFS.SOURCE TCS	
Description	FOR BROWSER CONNECTOR
Source Type	SESSION
Log Detail Level	NONE
Syntax Type	XML
Generic User	INPUTTER

OFS.SOURCE TAABS	
Description	FOR TAABS LOAD
Source Type	TELNET
Login Id.1	psgdev
Log Detail Level	NONE
Maint Msg Dets	Y
Det Prefix	IM
Syntax Type	OFS
Generic User	TAABS.INPUTTT

The Banking Software Company

Testing Telnet mode



- Type tSS OFS.source.id at the jshell prompt
- You should get a screen similar to the one shown below

```
jsh 200711 ~ -->tSS TAABS
<tSS version="1.1"><t24version>200711</t24version><t24pid>4840</t24pid><t24ofssource>TAABS</t24ofssource><clientIP/></tSS>
```

- Enter the following OFS enquiry request

ENQUIRY.SELECT,,AUTHOR/098765,CATEGORY-LIST

Response

```
""" 62267    "Oth Opertg Exp7"," 62268"    "Oth Opertg Exp8"," 62269" "
Oth Opertg Exp9"," 62270"    "Oth OpertgExp10"," 62999"    "Oth Oper Exps
"," 63000"    "Taxes      "," 63010"    "Federal Tax      "," 63020"
Local Tax     "," 63999"    "Taxes      "," 64000"    "Prov for Exps
"," 64499"    "Prov for Exps  "," 64500"    "Depreciation      "," 64505"
Depr on Land   "," 64529"    "Depr on Land   "," 64530"    "Depr on Bldgs
"," 64535"    "Depr Bldgs Own  "," 64540"    "DeprOther Buill"," 64541"
DeprOther Buil2"," 64542"    "DeprOther Buil3"," 64543"    "DeprOther Buil4
"," 64544"    "DeprOther Buil5"," 64599"    "Depr on Bldgs      "," 64600"
DeprMovbleAssts "," 64605"    "Depr Furn/Equip"," 64610"    "Depr IT Hardwar
"," 64615"    "Depr Software  "," 64620"    "Depr Vehicles      "," 64625"
Depr MovbleAs1 "," 64626"    "Depr MovbleAs2 "," 64627"    "Depr MovbleAs3
```

- Enter the following OFS transaction request

ABBREVIATION,/I/PROCESS,GERARD/654321,CUST,ORIGINAL.TEXT=CUSTOMER

Response

```
ABBREVIATION,/I/PROCESS,GERARD/654321,CUST,ORIGINAL.TEXT=CUSTOMER
CUST/IM072500009/1,ORIGINAL.TEXT=CUSTOMER:1:1,RECORD.STATUS=INAU:1:1,CURR.NO=1:
1:1,INPUTTER=19 THOMAS _OFS TAABS:1:1,DATE.TIME=0712171545:1:1,CO.CODE=GB001000
1:1:1,DEPT.CODE=1:1:1
```

- Create a TELNET type OFS.Source. Use this record to try out the following messages. Switch on OFS.SOURCE. Check the ORD entries after each request
- Create a transaction type request to carry out a Funds Transfer from one account to the other . Important fields are
 - TRANSACTION.TYPE
 - DEBIT.ACCT.NO
 - DEBIT.CURRENCY
 - DEBIT.VALUE.DATE
 - DEBIT.AMOUNT
 - DEBIT.CURRENCY
- Create another transaction request to authorise this funds transfer
- Use a enquiry type request to display the authorised transaction

- We will look at some special options next
 - How do we handle commas in a message?
 - How do we disable OFS for an application?

- Some OFS messages may contain commas as part of the data
 - An OFS request to create a version would use comma as part of the record id
 - Eg: A version MEMBER of CUSTOMER would have an id CUSTOMER, MEMBER
- Since comma is used as a delimiter in an OFS request, including a comma in the data portion would convey a completely different meaning
- Therefore use ? instead of commas in the data
- Questions marks are automatically converted to commas before processing by the OFS module

- To prevent update through OFS.
- This may be done by specifying the value ".NOFS" in the ADDITIONAL.INFO field of the PGM.FILE for the application to be blocked.

- Write a OFS transaction request to create a zero authorised version for ACCOUNT called NEWACCOUNT
- Test this out using the OFS online mode
- Modify the FUNDS.TRANSFER record within PGM.FILE to prevent users from carrying out FTs through OFS.
- Test this out using the online mode

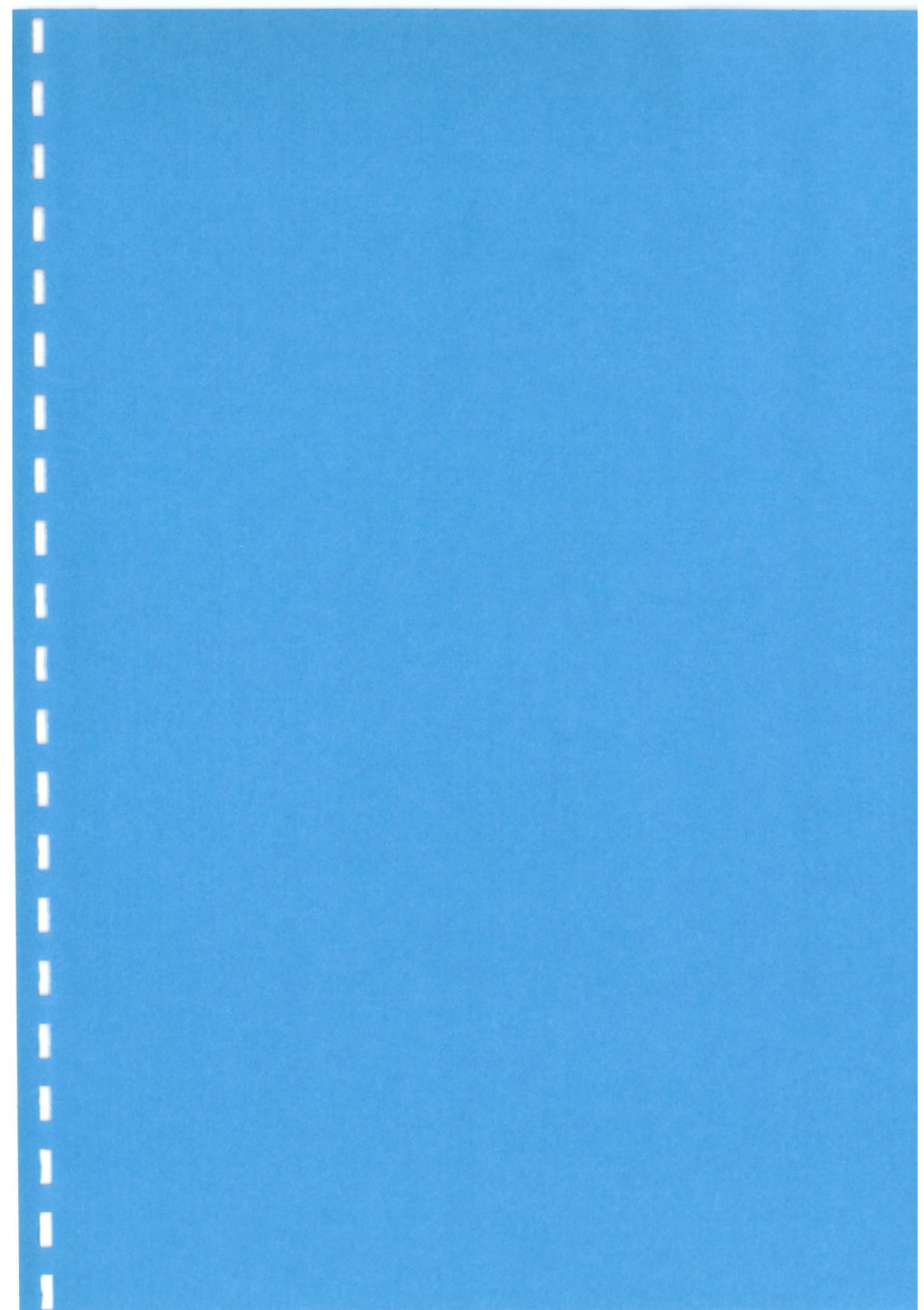
Thank You

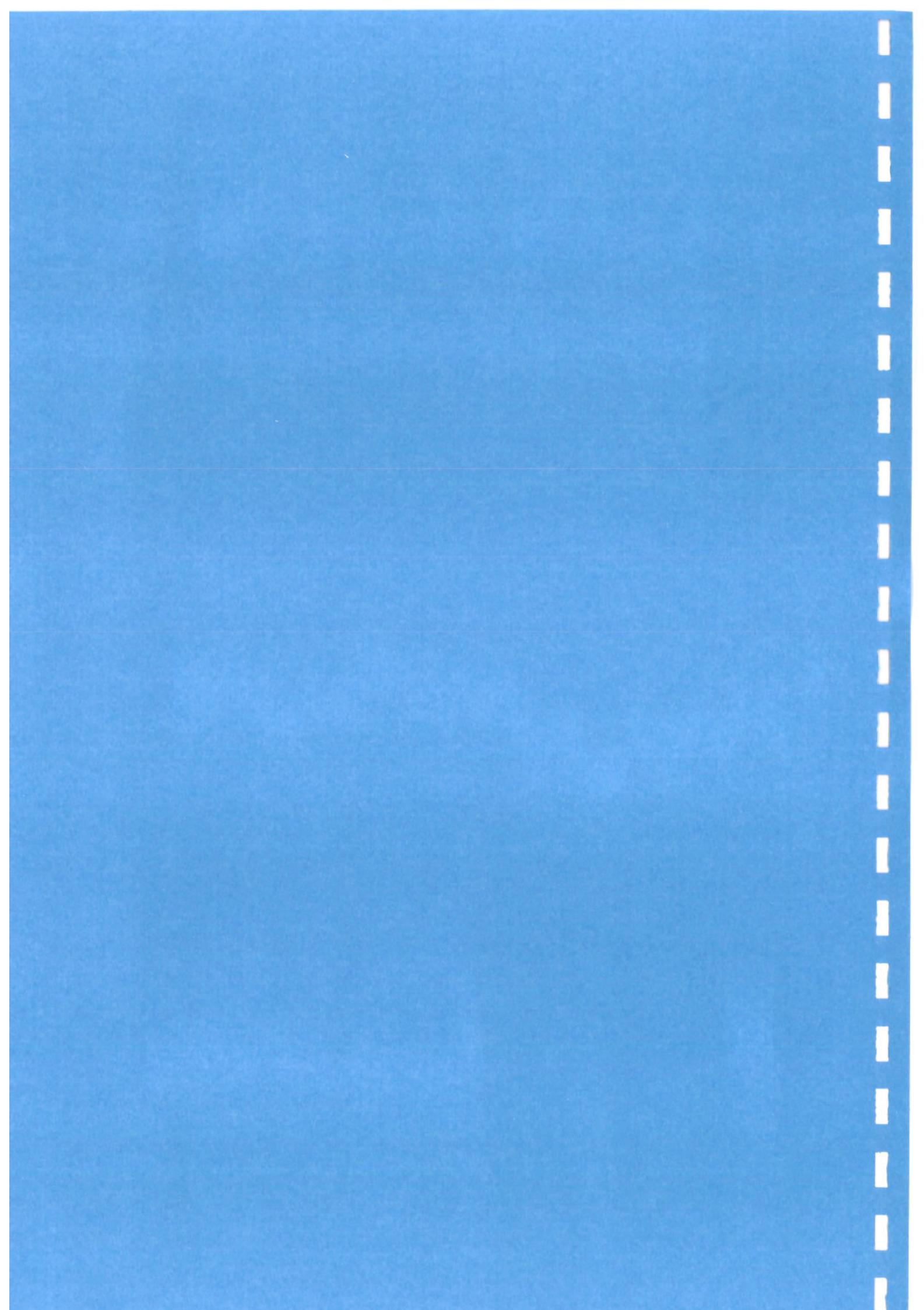
TEMENOS EDUCATION CENTRE

Warning: This document, is protected by copyright law and international treaties. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TEMENOS HEADQUARTERS SA. Unauthorized reproduction or distribution of this presentation or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under applicable law." Information in this document is subject to change without notice.



TEMENOS
The Banking Software Company





OFS Post Message

TEMENOS EDUCATION CENTRE

Warning: This document, is protected by copyright law and international treaties. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TEMENOS HEADQUARTERS SA. Unauthorized reproduction or distribution of this presentation or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under applicable law." Information in this document is subject to change without notice



TEMENOS

The Banking Software Company



The Banking Software Company

At the end of the session you will have

- Adequate knowledge to understand
 - Inter application OFS calls
- Adequate knowledge to work on
 - OFS.POST.MESSAGE
 - OFS.MESSAGE.SERVICE
 - DE.ADDRESS

- Previously done using OFS.GLOBUS.MANAGER
- Calls to OFS.GLOBUS.MANAGER from local APIs (eg: versions) deprecated from R5
- How do we call a OFS message from a version then?
- Use the OFS.POST.MESSAGE routine

- Syntax:

```
CALL OFS.POST.MESSAGE(Y.OFS.MESSAGE, OFS.MSG.ID, OFS.SOURCE.ID, OPTIONS)
```

- Explanation:

- Application name, record id and field information's will be passed through the first argument Y.OFS.MESSAGE

- OFS.SOURCE.ID must be the id of a record in OFS.SOURCE with type GLOBUS

- The supplied OFS message will get written on to a queue(a file in this case) named F.OFS.MESSAGE.QUEUE.
- OFS.MESSAGE.SERVICE picks up the data and processes.
- Once the message is processed, the response is written on to an out queue named F.OFS.RESPONSE.QUEUE.
- Message in OFS.RESPONSE.QUEUE may be deleted using its corresponding service OFS.RESPONSE.SERVICE

The Banking Software Company

- This is a standard T24 service
- It has its own workload profile , where the no of agents are usually set to 1.

OF5 MESSAGE SERVICE

TEMENOS

TSA.SERVICE
BIK-OF5.MESSAGE-SERVICE

Description	TSA OF5 message service
Server Name.1	OF5.MESSAGE-SERVICE
Work Profile.1	OF5.TSA.message.service
User	* INPUTTER INPUTTER
Service Control	STOP
Review Time	
Time Out	
Attribute Type.1	
Attribute Value.1	
Reserved 3	
Reserved 2	
Reserved 1	
Local Ref	
Record Status	
Curr No	1
Inputter.1	26_MARKOVERS
Date Time.1	12 DEC 06 10:37
Authoriser	26_MEHMETOKAN2
Co Code	DE0010001 C02 TESTBASE
Dept Code	
Auditor Code	
Audit Date Time	1

The Banking Software Company

TSA WORKLOAD PROFILE record

TEMENOS

TSA.WORKLOAD.PROFILE OFS.MESSAGE.SERVICE	
Description	OFS TSA message service
Time.1	4
Agents Required.1	1
Reserved 9	
Reserved 8	
Reserved 7	
Reserved 6	
Reserved 5	
Reserved 4	
Reserved 3	
Reserved 2	
Reserved 1	
Local Ref	
Record Status	
Curr No	1
Inputter.1	1_000512m
Date Time.1	12 DEC 05 10:29
Authoriser	26_MARKOVER1
Co Code	US0010001 BNK TESTSASE
Dept Code	1

The Banking Software Company

- DEADDRESS is normally updated with Customer address when a new customer record is authorised
- Automate the process of creating a second address in the DE.ADDRESS file when a Customer record is authorised

- Check if the customer record being authorized is a new customer record
- Get the value for the second address through a local reference field.
 - Use LOCAL.TABLE & LOCAL.REF.TABLE for local ref field
- Form a record for the DE.ADDRESS file with the short name, name.1, name.2 and street.address. The first 3 fields should be extracted from the corresponding fields in the customer record and street.address from the local reference field.
- Form the id of the DE.ADDRESS file
- Write the record formed into the DE.ADDRESS file using the routine OFS.POST.MESSAGE

- DE.ADDRESS records have ids usually created in the following format
 - company-id.C-customer-id.PRINT.1
- We need to create an id in the format
 - Companycode.C-CustomerNo.PRINT.2
 - Eg: US0010001.C-100069.PRINT.2
- Get the second address from the customer through a local reference field.
- How do we get the current company id?

- This is a dynamic array that is defined in the I_COMMON file which holds the id of the current company.

```
SUBROUTINE V.TRG.AUTH.RTN
$INSERT I_COMMON
$INSERT I_EQUIATE
$INSERT I_F.CUSTOMER
$INSERT I_F.DE.ADDRESS

GOSUB INIT
GOSUB OPENFILES
GOSUB PROCESS
RETURN

INIT:
FN.CUS = 'F.CUSTOMER'
F.CUS = ''
Y.ADDRESS = ''
RETURN

OPENFILES:
CALL OPF(FN.CUS,F.CUS)
RETURN
```

```
PROCESS:  
    IF R.OLD(1) = '' THEN      /* If it is a new record being  
authorised  
  
        * Form the id of the DE.ADDRESS record  
  
        Y.DE.ADDRESS.ID = ID.COMPANY:'C-' : ID.NEW:'PRINT.2'  
        Y.ADDRESS = R.NEW(EB.CUS.LOCAL.REF)<1,7>  
  
Y.OFS.REC="DE.ADDRESS,,INPUTT/654321," : Y.DE.ADDRESS.ID:",SHORT.NAME=":  
R.NEW(EB.CUS.SHORT.NAME):",NAME.1=":R.NEW(EB.CUS.NAME.1):",NAME.2=":R.  
NEW(EB.CUS.NAME.2):", STREET.ADDR=:Y.ADDRESS  
  
OFS.MSG.ID=""  
OFS.SOURCE.ID="TRG.OFS.GLOBUS"  
OPTIONS=""  
  
CALL OFS.POST.MESSAGE (Y.OFS.REC, OFS.MSG.ID, OFS.SOURCE.ID,  
OPTIONS)  
  
END  
  
RETURN  
END
```

The Banking Software Company

- Create and add a local reference field called ADDRESS2 to CUSTOMER
- Create a version of the Customer application with all the mandatory fields and the local ref field ADDRESS2
- Compile and catalogue the subroutine and attach it to the Auth.Rtn field of this version.
- Enter a new customer record using the version
- Authorise the record

- Check if OFS.MESSAGE.QUEUE has been updated
- Start the following services
 - TSM
 - OFS.MESSAGE.SERVICE
- Check if the message has been removed from OFS.MESSAGE.QUEUE
- Check if you have the response in OFS.RESPONSE.QUEUE

- Inter application OFS calls was previously done using OFS.GLOBUS.MANAGER
- We use OFS.POST.MESSAGE routine to call an OFS message from a version.
- OFS.MESSAGE.SERVICE picks up the data and processes the same.

Thank You

TEMENOS EDUCATION CENTRE

Warning: This document, is protected by copyright law and international treaties. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TEMENOS HEADQUARTERS SA. Unauthorized reproduction or distribution of this presentation or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under applicable law." Information in this document is subject to change without notice



TEMENOS
The Banking Software Company