TEMENOS™

TEMENOS T24

OFS

User Guide

Table of Contents

Overview

OFS (Open Financial Service) is an interface that allows transactions and queries requests to be processed by T24.

This document describes the workings, features and syntax of OFS itself, and does not describe the connectivity options available for communicating with OFS. This information may be found in other chapters of the T24 Connectivity User Guides, and XML schema information may be found at www.temenos.com and http://dts.temenos.com.

OFS is the single point of entry to T24. Every single interaction with T24 is driven through OFS.

OFS offers the following distinct request types:

- Transactions
- Enquiries
- Clearing
- XML report requests

And offers the following features:

- Audit trail of messages (optional)
- API hooks for pre and post processing
- Fully multi company / multi book aware
- Full support for hierarchical data (multi values, sub values)
- Full multi-lingual support

Conventions Used in this Document

Throughout this document, examples are shown with the USER sign on name of TEST.USER and a password of 654321.

OFS message examples use text in `Courier new.`

Example OFS Requests

Transaction Request

```
ABBREVIATION,,TEST.USER/654321,SEC,ORIGINAL.TEXT=SECTOR

ABBREVIATION,/I,TEST.USER/654321,SEC,ORIGINAL.TEXT=SECTOR

ABBREVIATION,,TEST.USER/654321/DE0010001,SEC,ORIGINAL.TEXT=SECTOR

ABBREVIATION,//VALIDATE,TEST.USER/654321,SEC,ORIGINAL.TEXT=SECTOR

ABBREVIATION,OFS.VERSION,TEST.USER/654321,SEC,ORIGINAL.TEXT=SECTOR

ABBREVIATION,OFS.VERSION //VALIDATE,TEST.USER/654321,SEC,ORIGINAL.TEXT=SECTOR2

ABBREVIATION,OFS.VERSION /R,TEST.USER/654321,SEC

ABBREVIATION,/D,TEST.USER/654321,SEC

HELPTEXT.MENU,/D,TEST.USER/654321,OFS.TEST

HELPTEXT.MENU,,TEST.USER/654321,OFS.TEST,APPLICATION:1:=SECTOR,DESCRIPT:1:=Sector

HELPTEXT.MENU,OFS.DEMO/I/PROCESS,TEST.USER/654321,OFS.TEST,APPLICATION:1:=SECTOR,DESCRIPT:1:=S
ector,APPLICATION:2:=INDUSTRY,DESCRIPT:2:=Industry,DESCRIPT:2:2=Industrie
```

The following example relates to a foreign exchange swap transaction where information is required for both legs of the swap. The _ (underscore) character is used to delimit information for record 1 from record 2:

```
FOREX,GTS//PROCESS,GTUSER/QWERTY,,DEAL.TYPE=SW,COUNTERPARTY::=100076,CURRENCY.BOUGHT::=USD,AMO
UNT.BOUGHT::=500000,CURRENCY.SOLD::=GBP,SPOT.RATE::=1.6,DEAL.DATE::=20011026,VALUE.DATE.BUY::=
20011028_1M,FORWARD.RATE::=_1.55,REVALUATION.TYPE:_1:=_IN,NOTES:1_1="first        leg"_"second
leg",NOTES:_2:_"second-line":VALUE.DATE.BUY::=20011028_1M,
```

Enquiry Request

```
ENQUIRY.SELECT,,TEST.USER/654321,CURRENCY-LIST

ENQUIRY.SELECT,,TEST.USER/654321,CURRENCY-LIST,@ID:LK=C...
```

XML Report Request

```
XML.REPORT,ID,TEST.USER/654321,XML.CURRENCY.LIST

XML.REPORT,XML,TEST.USER/654321,XML.CURRENCY.LIST
```

TEC

```
TEC.OFS.INTERFACE,,TEST.USER/654321,,FLUSH

TEC.OFS.INTERFACE,,TEST.USER/654321,,API.RESPONSE,MY.API,VERSION ROUTINE,1234
```

Clearing Request

```
CLEARING,,TEST.USER/654321,OFS.DEMO,TXNREF1,33537,USD,10000,C,SALARY_TXNREF2,32549,USD,8000,C,
SALARY

CLEARING,,TEST.USER/654321,OFS.DEMO,TXNREF1,33537,USD,10000,C,SALARY_,32549,USD,8000,C,SALARY
```

Setup

Before OFS can be used, some simple configuration of *OFS.SOURCE* is required. Refer to the appendix for details of the fields that are on *OFS.SOURCE* and their use.

NB this deals only with the configuration of OFS itself. Details on the configuration of the connectivity layer (also referred to as "the connectors" or "TOCF") can be found in the user guide chapter "T24 Temenos Connector".

Request Syntax Overview

OFS uses a comma delimited message syntax. Here, the components of the message are colour coded so that it is easier to identify the components in the following examples:

OPERATION,OPTIONS,USER INFORMATION,ID INFORMATION,DATA

OPERATION

Indicates the operation to perform, being one of:

- ENQUIRY.SELECT – runs an enquiry
- CLEARING – invokes clearing (OFS.CLEARING.MANAGER)
- XML.REPORT – runs an XML report
- T24 application – transaction request

USER INFORMATION

Contains the details of the T24 user that will execute the transaction passed to T24, including sign on name, password and company code delimited by the / character in the following syntax:

USER.NAME/PASSWORD/COMPANY

The company is optional. Hence both of the below examples are valid:

TEST.USER/654321/GB0010001

TEST.USER/654321

When no company is specified, the default company for the user will be used.

Replace option

There is a special option which can be used but with due care, that will allow records to be cleared and re-input. For example replacing a multi-value set containing 5 records with a new set containing only 3.

If used the preceding / characters need to be included.

TEST.USER/654321/GB0010001///1

TEST.USER/654321////1

OPTIONS, ID INFORMATION and DATA

These are specific to the type of request. Refer to the following sections for more detail.

Transactions

In this section we refer to the following message:

```
HELPTEXT.MENU,OFS.DEMO/I/PROCESS,TEST.USER/654321,OFS.TEST,APPLICATION:1:=S
ECTOR,DESCRIPT:1:=Sector,APPLICATION:2:=INDUSTRY,DESCRIPT:2:=Industry,DESCR
IPT:2:2=Industrie
```

Operation (Blue)

Any valid T24 application or abbreviation is permitted.

Options (Red)

For application requests, the options may specify a VERSION to be used, the FUNCTION to be used and a processing flag of either PROCESS or VALIDATE. The options section is delimited by the / character.

VERSION/FUNCTION/PROCESSING FLAG

Where no FUNCTION is specified, it defaults to I. Where no processing flag is specified, it defaults to PROCESS. VALIDATE indicates the all normal processing takes place without actually updating the database, an example use of which would be to return all default fields. PROCESS indicates that the message is to be committed to the database.

Below are some examples of using options and the effect:

Options used	VERSION	FUNCTION	FLAG
OFS.DEMO/I/PROCESS	OFS.DEMO	I	PROCESS
OFS.DEMO	OFS.DEMO	I	PROCESS
OFS.DEMO/I/VALIDATE	OFS.DEMO	I	VALIDATE
/R	Not set.	R	PROCESS
/D	Not set.	D	PROCESS
OFS.DEMO/V	OFS.DEMO	V	PROCESS
//VALIDATE	Not set.	I	VALIDATE

NB Care should be taken when using the V function in OFS as this may cause long running processes to be invoked.

ID Information (Yellow)

Specifies the transaction reference to be used, and must be a valid key to the application being invoked. If this is left blank, this can be automatically allocated for new transactions by those applications that support this feature.

```
TRANSACTION.ID/MESSAGE.ID
TRANSACTION.ID
```

If specified, the id of the message being currently processed can be passed in.

Data (Black)

`FieldName:MultiValueNumber:SubValueNumber=Content`

Contains the data required to create/update the transaction, this is repeated for as many data items as required, each item being separated by a ',' where:

FieldName must be defined in the *STANDARD.SELECTION* record for the application. At least one field must be specified where the function is I.

- MultiValueNumber targets a specific multi value. If not set, then the first multi value is assumed. Not required for non – multi value fields.
- SubValueNumber targets a specific sub value. If not set, then the first sub value is assumed. Not required for non – sub value fields.
- Content defines the data to be set to the specified field will be validated as per the standard application.

If 'NULL' is specified as field data, OFS will blank the field of all data. This should not be used to remove multi-values or sub values. As with any data entry / modification in OFS, the validation rules must be observed. Should a mandatory entry field be set to NULL then an error would be generated.

To remove multi-value or sub value fields a minus sign (-) should be entered into the field data. Only the first field needs be specified as a minus to remove a sub or multi value set.

Data Item	Field Name	Multi Value	Sub Value	Data
DISPL.APPLICATION=Y	DISP.APPLICATION	n/a	n/a	Y
DISPL.APPLICATION::=Y	DISP.APPLICATION	n/a	n/a	Y
APPLICATION::=SECTOR	APPLICATION	1	n/a	SECTOR
APPLICATION:2:=INDUSTRY	APPLICATION	2	n/a	INDUSTRY
DESCRIPT:2:=Industry	DESCRIPT	2	1	Industry
DESCRIPT:2:2=Industrie	DESCRIPT	2	2	Industrie

Note:

If a ',' (comma) character is required within the field content then it should be replaced by the '?' (Question mark) character.

Using VERSION to control application requests

Each transaction to be updated using OFS can be configured using the standard *VERSION* utility. This can be used to define default values, and can be customised with user-defined sub-routines.

The version will also control the handling of error messages and override conditions. The following options are available:

- Reject message on error
- Place new transactions in HLD status on error/override condition
- Automatically approve override conditions encountered.
- Automatically place all transactions on hold.

The *VERSION* field controlling these options is `GTS.CONTROL`

NAU processing

There is a facility on *VERSION* to specify optional functionality for the processing of transactions via OFS where $NAU records exist. The field `NAU.PROCESSING` in *VERSION* can be set according to the type of NAU processing required.

0	Reject messages.
1	Overlay message.
2	Accept Reversal as Deletion.
3	Apply both option 1 and option 2 to this version.

Refer to the section Processing NAU Records in *VERSION* chapter for a detailed explanation of this feature.

Non Standard Applications

Foreign Exchange Swaps

Foreign exchange swaps require that information is provided for both legs of a swap within a single message. In this case information for the first leg is separated from information for the second leg by a '-' (underscore) character. This applies to the multi value number, sub value number, and field data parts of the message.

Loans and Deposit contracts with Payment Schedules

An LD contract with a payment schedule requires the information for the linked application *LD.SCHEDULE.DEFINE*. This can be done by adding the field information for *LD.SCHEDULE.DEFINE* to the Loans and deposits MESSAGE DATA component, separated by '//' (2 forward slashes).

When using OFS to update *LD.LOANS.AND.DEPOSITS* with *LD.SCHEDULE.DEFINE*, the `GTS.CONTROL` field on both versions must be set to 1, overrides and accepted and errors placed on hold.

Enquiries

```
ENQUIRY.SELECT,,TEST.USER/654321,CURRENCY-LIST,@ID:LK=C...
```

Operation (Blue)

ENQUIRY.SELECT

To run an ENQUIRY, the operation must be set to ENQUIRY.SELECT.

Options (Red)

There are no options used for enquiries.

ID Information (Yellow)

ENQUIRY.ID

The ID to the ENQUIRY to run. Mandatory.

Data (Black)

SELECTION.FIELD:OPERAND=CRITERIA

The selection criteria with which to run the enquiry (optional) where:

SELECTION.FIELD is the name of the field to select and must be either a valid SELECTION.FLDS for the Enquiry or defined in the *STANDARD.SELECTION* record for the FILE.NAME of the Enquiry.

- OPERAND is the operand for the selection, must be specified with a SELECTION.FLDS, may be EQ, NE, GE, GT, LE, LT, UL, LK, and NR.
- CRITERIA is the data value for the SELECTION.FLDS and requested operand for the selection

Each Message Data specification should be separated by a ‘,’.

Clearing

```
CLEARING,US0010001,TEST.USER/654321,PAY,123456789,18899,USD,10
000,C,NET SALARY_ PAY,7654321,23884,USD,20000,D,NET SALARY
```

Operation (Blue)

CLEARING

To process a query request, the operation must be set to CLEARING

Options (Red)

A company code may be specified and is used as the company that the entries are raised in.

ID Information (Yellow)

AC.ENTRY.PARAM.ID

The ID to the AC.ENTRY.PARAM record that is used to define the layout of the data section.

Data (Black)

Each part of the entry should be separated by the character defined in the `FIELD.DELIM` field of the *AC.ENTRY.PARAM* record. It is recommended that a comma is used.

Each entry is separated by a '_'

Configuration

The layout of the data section of OFS Clearing is controlled using *AC.ENTRY.PARAM*. For further details on configuring *AC.ENTRY.PARAM*, refer to the section on the Generic Accounting Interface in the Local Clearing User Guide.

XML reports

```
XML.REPORT,XML,TEST.USER/654321,PXML.CURRENCY.LIST
```

Operation (Blue)

```
XML.REPORT
```

To run an xml report, the operation must be set to XML.REPORT

Options (Red)

Either XML or ID. ID returns the *HOLD.CONTROL* key that was produced when the report was run, such that the XML can be extracted asynchronously. XML will return the XML result.

ID Information (Yellow)

```
REPORT.CONTROLID
```

The key of the ENQUIRY.REPORT to run.

Data (Black)

Not required. It is possible to verify *ENQUIRY* records and have the &HOLD& ID or the entire XML message brought back in the return message.

The application title has changed from *ENQUIRY.REPORT* to XML.REPORT, the OFS module will translate the XML.REPORT script and apply it to the application *ENQUIRY.REPORT*.

Configuration

XML reports are configured via *ENQUIRY.REPORT*, where the `OUTPUT.FORMAT` should be set to XML and the corresponding *REPORT.CONTROL* record has the `FORM.NAME` set to HOLD. Refer to the Enquiry user guide for more detail.

TEC OFS Interface- Request

TEC.OFS.INTERFACE,,TEST.USER/654321,,API.RESPONSE,MY.API,VERSION ROUTINE,1234

TEC.OFS.INTERFACE,,TEST.USER/654321,,API.RESPONSE,MY.API,VERSION ROUTINE,1234

Operation (Blue)

TEC.OFS.INTERFACE

To invoke the TEC OFS interface, the operation must be set to TEC.OFS.INTERFACE

Options (Red)

Not applicable.

ID Information (Yellow)

Not applicable.

Data (Black)

The data passed is comma delimited and composed of:

ITEM.ID,MY.KEY,MY.DETAIL,MY.VALUE

- ITEM.ID Key to TEC.ITEMS
- MY.KEY The key of the data passed
- MY.DETAIL Detail of the item
- MY.VALUE Any value (size, time, etc.) associated with the item. Optional

Configuration

Refer to the TEC user Guide for details on configuring the T24 Enterprise Console.

Response Syntax Overview

OFS uses a comma delimited response syntax. Each request type has a different response syntax detailed below.

Transactions

Overview

The response syntax for a transaction request is returned in the following format:

TRANSACTION ID/MESSAGE ID/SUCCESS INDICATOR,RESPONSE DATA

```
SEC//1,ORIGINAL.TEXT=SECTOR:1:1,RECORD.STATUS=INAU:1:1,CURR.NO=2:1:1,INPUTTER=11_TONY1___OFS_T
CS:1:1,DATE.TIME=0622091548:1:1,CO.CODE=US0010001:1:1,DEPT.CODE=1:1:1
```

Transaction ID

This will either contain the id supplied initially, or if no id was supplied the next available id (if generated).

Message ID

Contains the OFS message reference of the processed message.

Success Indicator

This flag will indicate the success or failure of the request for both application and enquiry transactions. The following values can be returned:

1	Successful transaction
-1	Error encountered
-2	Override condition encountered
-3	System offline

Response Data

A successfully processed update will return the fully populated record as a repeating string separated by a comma in the same format as the data passed to OFS:

`FieldName:MultiValueNumber:SubValueNumber=Content`

A transaction encountering errors will return the list of fields that were in error, separated by a comma, in the following formant:

`FieldName:MultiValueNumber:SubValueNumber=ErrorMessage`

Where:

- FieldName must be defined in the *STANDARD.SELECTION* record for the application. At least one field must be specified where the function is I.
- MultiValueNumber targets a specific multi value. If not set, then the first multi value is assumed. Not required for non – multi value fields.
- SubValueNumber targets a specific sub value. If not set, then the first sub value is assumed. Not required for non – sub value fields.
- Content defines the data to be set to the specified field will be validated as per the standard application.
- ErrorMessage contains error message generated for the associated Field name, Multi Value, sub value. The message will be returned with variable elements expanded.

Enquiries

Overview

The data returned from enquiry will be in the following format, in three sections:

```
HEADER CAPTION DETAILS,COLUMN DETAILS,RESPONSE DATA

,@ID::Key/CCY.NAME::Name,"CAD"  "CANADIAN DOLLAR","CHF" "SWISS FRANCS"
```

Header Caption Details

```
Identifier1=Text1/Identifier2=Text2
```

Where:

- Identifier - contains an identifier to determine which element of the header / caption is being defined. May be alphanumeric defined in the underlying *ENQUIRY*.
- Text - Contains the text for the corresponding identifier.

Each repeating series of header definitions is delimited by a '/'.

Column Details

```
Identifier1:format type1:label1/Identifier2:format type2:label2
@ID::Key/CCY.NAME::Name
```

Where :

- Identifier - contains the column identifier, which can be a name or a number. Each column to return information must be defined in the underlying *ENQUIRY*.
- Format type - contains the type of data contained in the column. This information can be used for formatting. Possible types include DATE (formatted using standard date formats) and AMOUNT (formatted to an amount with decimal format).
- Label - indicates the name of the column used as the column header.

Each series of column details is followed by a '/'

Response Data

```
Row value <TAB>
"CAD"  "CANADIAN DOLLAR","CHF" "SWISS FRANCS"
```

Each row will be comprised of a number of columns defined in the previous element of the message data. For each column the value will be returned delimited by the <tab> character. Each row is delimited by a ‘,’.

The returned fields are separated by the tab (ASCII 9) character with lines separated by ‘,’ (comma).

Clearing

Currently the Clearing interface does not give a response for a successful transaction.

In the case of an error, the response contains the error message.

XML Reports

Overview

REPORT ID/MESSAGE REFERENCE/SUCCESS INDICATOR/RESPONSE DATA

XML.CURRENCY.LIST//1/14145000115984801

XML.CURRENCY.LIST //1/<T24>…</T24>

Report ID

The ID of the report that was run.

Message Reference

The OFS message reference, where applicable.

Success Indicator

1 indicates the request was processed successfully. -1 indicates that an error was encountered, in which case the Response Data holds the error message

Response Data

Contains either the key to the &HOLD& table, e.g. 137740002641133401, where the contents of the report has been stored, or the actual XML response itself. An example of the XML output is shown below in “pretty print” format. For further information refer to the OFSML schema documentation.

```
<T24 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.temenos.com/T24/OFSML/130"
xsi:schemaLocation="http://www.temenos.com/T24/OFSML/130 ofsml.xsd">
 <ofsmlHeader>
  <requestId>0001159949.01</requestId>
  <requestTimeStamp>2006-SEP-22T16:39:09Z</requestTimeStamp>
  <requestExpiryTime>P0Y0M1DT0H0M0S</requestExpiryTime>
</ofsmlHeader>
<serviceResponse>
 <ofsReport docType="XML.REPORT">
  <dataSet name="CURRENCY-LIST">
   <fieldDefinition index="1" id="@ID" label="Key" type="ALPHANUMERIC" length="3"
repeatable="FALSE"/>
   <fieldDefinition index="2" id="CCY.NAME" label="Name" type="ALPHANUMERIC" length="35"
repeatable="TRUE"/>
   <record>
     <field index="1">CAD</field>
     <field index="2">CANADIAN DOLLAR</field>
   </record>
   <record>
     <field index="1">CHF</field>
     <field index="2">SWISS FRANCS</field>
   </record>
  </dataSet>
 </ofsReport>
</serviceResponse>
</T24>
```

TEC OFS Interface- Response

Overview

```
SUCCESS INDICATOR,RESPONSE DATA

1,ITEM.ID=API.RESPONSE,KEY=MY.API,DETAIL=VERSION ROUTINE,VALUE=1234
```

Success Indicator

1 indicates the request was processed successfully. -1 indicates that an error was encountered, in which case the Response Data holds the error message

Response Data

Confirms the data that was sent in the request:

```
ITEM.ID=PASSED.ITEM,KEY=PASSED.KEY,DETAIL=PASSED.DETAIL,VALUE=PASSED.VALUE
```

Other Features

Multi-company processing

OFS will allow transactions to be processed in any company that the user has access. The company for the transaction can be specified as part of the message syntax.

Message Logging

A logging facility is provided for all messages for a particular service. This can log all traffic, exception messages, communication start and end or no messages. Full history of each message can be recorded if required. The details are stored in the *OFS.REQUEST.DETAIL* file if message audit is required. If message auditing is required, full details of the message and subsequent processing statuses will be recorded. Message logging is configured in *OFS.SOURCE*.

API Hooks

Processing can be customised to suit specific requirements by the use of API at various points within the processing path. This is configured in *OFS.SOURCE*. User routines can be incorporated at the following points:

Communication Start and End

This will allow specific operations to take place at the start and end of the communication process.

Individual message receipts and returns

This will allow manipulation of the message immediately after receipt and immediately after return to/from the calling application.

Individual message pre and post process

This allows the message to be manipulated prior to and on return from processing.

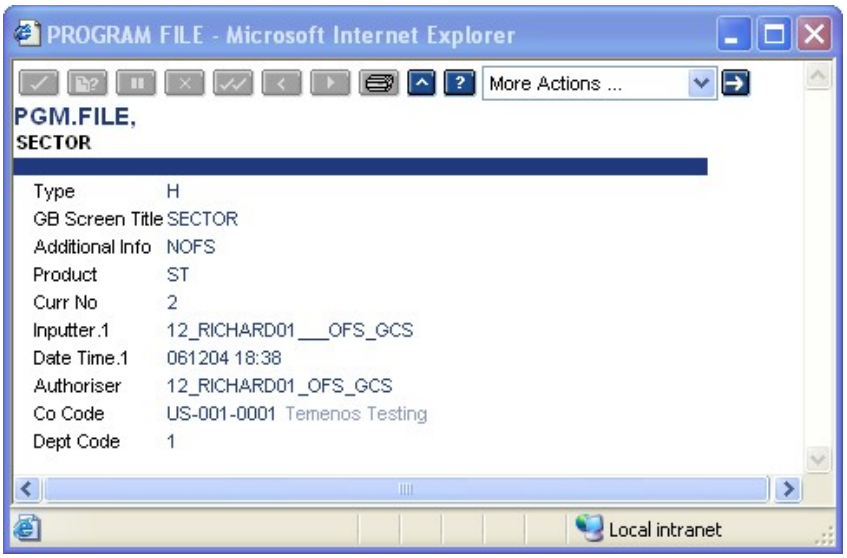
Security and OFS

In order to execute an OFS transaction a sign on name and password must be supplied by the source application. This will be validated before processing any online request.

Note: A valid sign on and password must be supplied before the online server can accept a message.

Once validated all access to data is controlled using the standard SMS definitions specified in the user profile.

It is possible to block applications from being used with OFS by any user. This may be done by adding NOFS to the `ADDITIONAL.INFO` field of the *PGM.FILE* for the application to be blocked.



PGM.FILE record with NOFS signalled in the ADDITIONAL.INFO field

Note: As the *PGM.FILE* application controls various aspects of the running of applications; changes should only be made with the utmost caution. While adding .NOFS to the `ADDITIONAL.INFO` field will simply prevent the use of OFS on that application, other changes could damage the integrity of the system.

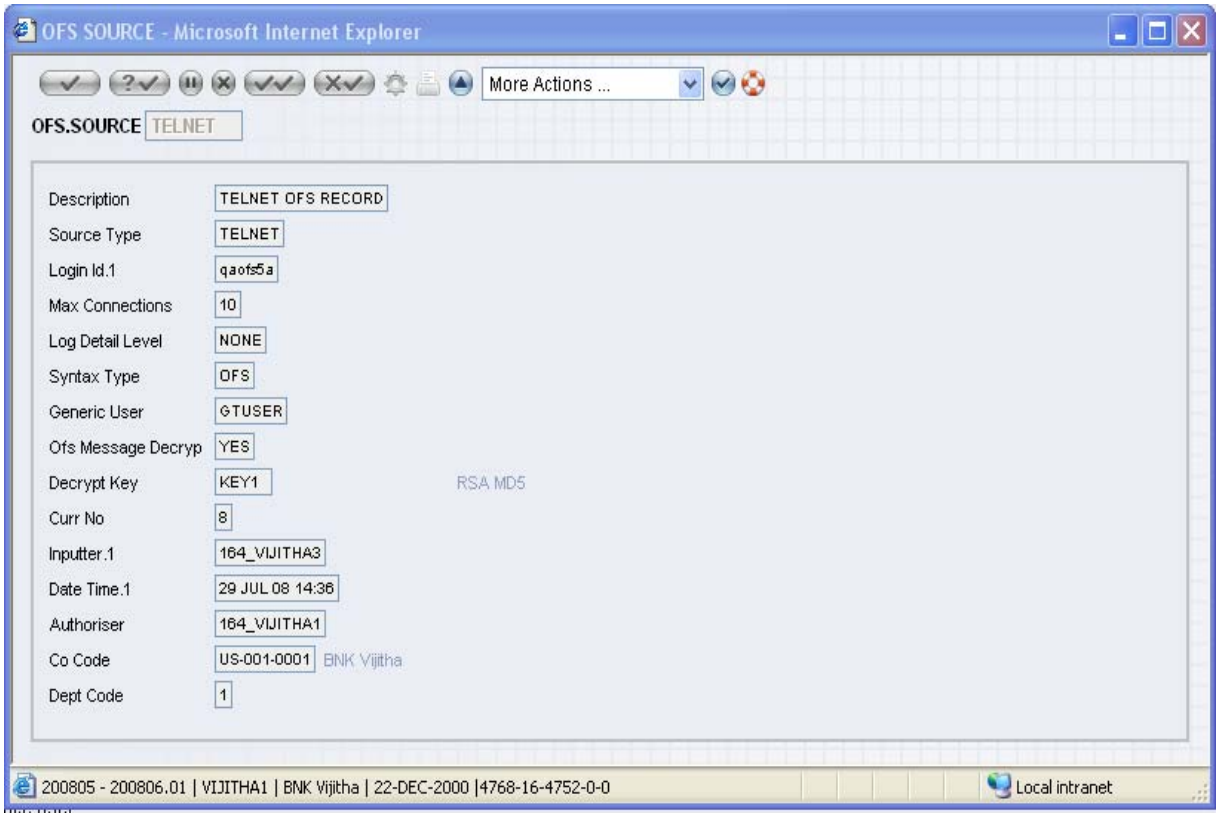
Encryption of Message

OFS.SOURCE

There are two fields in OFS.SOURCE which handle encryption: OFS.MESSAGE.DECRYPT & DECRYPT.KEY

OFS.MESSAGE.DECRYPT specifies whether Decryption should take place or not.

If OFS.MESSAGE.DECRYPT is set to 'Yes', then a valid record in OFS.DECRYPT.KEY is required which contains the necessary information for Decryption.

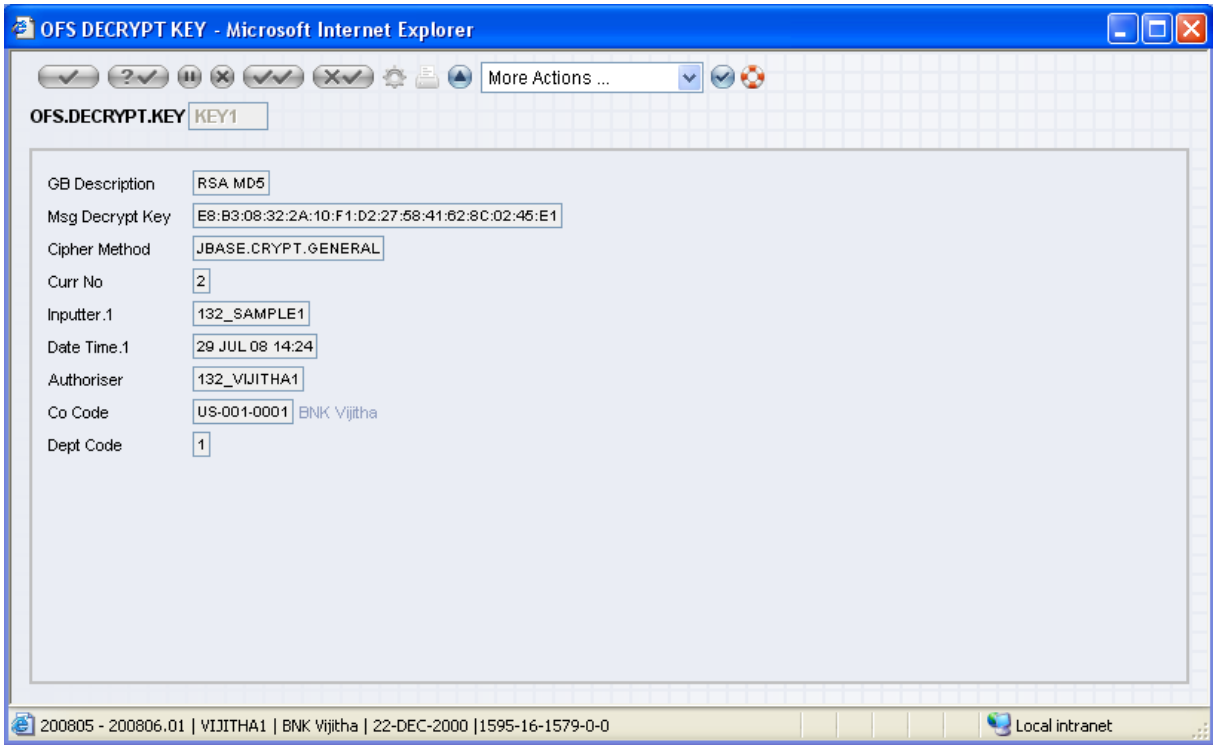


OFS.SOURCE with decryption set

OFS.DECRYPT.KEY

The MSG.DECRYPT.KEY field contains the key with which the decryption will take place. The field CIPHER.METHOD holds the desired Decryption method.

The options for CIPHER.METHOD are provided by a dropdown list.



OFS.DECRYPT.KEY

ENCRYPTION and DECRYPTION

The facility of Encrypting and Decrypting a message in JBASE is handled by using the ENCRYPT and DECRYPT commands respectively.

Wrapping Messages with Tags

It is recommended that messages are wrapped with and opening and closing set of matching tags to clearly identify where each message begins and ends which may be necessary when processing large messages.

This is done by adding to the start of a message a tag packaged between < > signs and repeating the tag packaged between </ >. The '/' sign indicates the end of the message, c.f. XML tags.

Example:

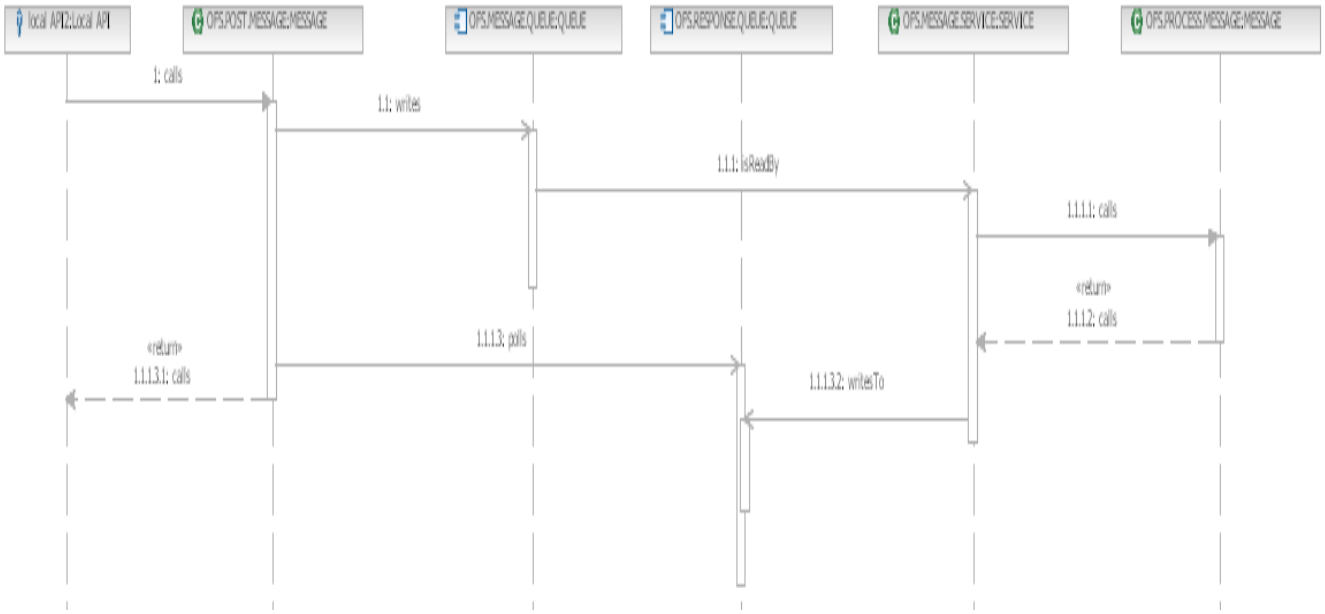
```
<MSG1>ABBREVIATION,INPUTT/******,SEC,ORIGINAL.TEXT=SECTOR</MSG1>
```

In the example above the confirmation message returned will also be prefixed with <MSG1> and suffixed with </MSG1>.

Using OFS from API Code

Do not use OFS.GLOBUS.MANAGER, OFS.REQUEST.MANAGER or OFS.PROCESSOR.MANAGER directly. These routines are internal routines and do NOT form part of the T24 public API (and transaction boundaries will become corrupt if these routines are invoked directly). Instead, the routine OFS.POST.MESSAGE should be used.

A TSA service has been created to process OFS messages from outside a transaction.



The local API calls OFS.POST.MESSAGE, passing the *OFS.SOURCE* record to use and one or many OFS messages (delimited by a value mark - VM). The response is the key to the OFS.MESSAGE.QUEUE table.

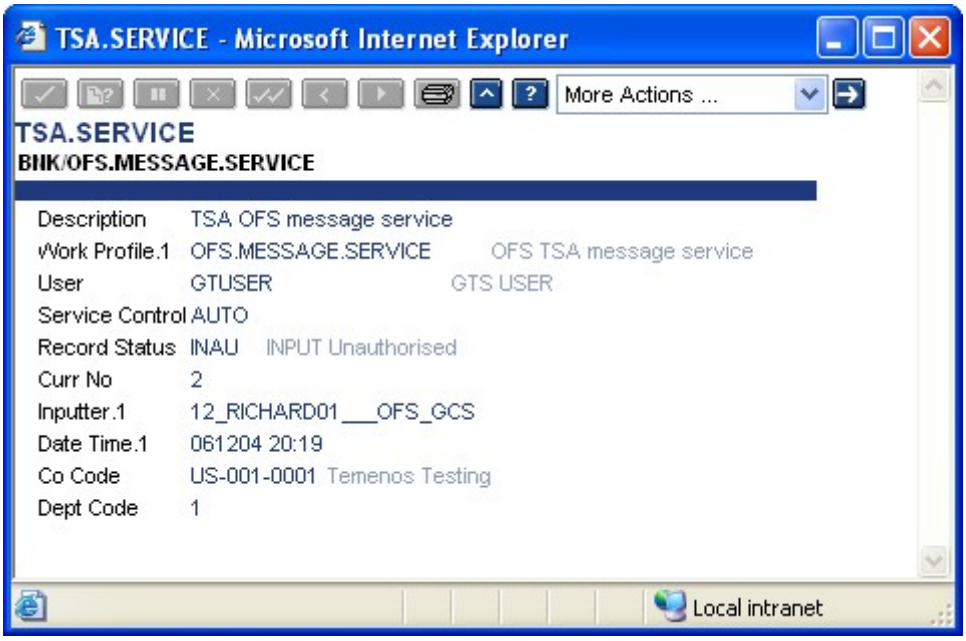
OFS.POST.MESSAGE writes the request out the to the OFS.MESSAGE.QUEUE table, which is the trigger table for OFS.MESSAGE.SERVICE. The service picks up the message and processes through OFS.PROCESS.MANAGER. Once processed, the record is removed from OFS.MESSAGE.QUEUE and posted to OFS.RESPONSE.QUEUE with the same key with OFS response success/fail flag in the first field and the actual OFS response in the 2nd field.

A second service, OFS.RESPONSE.QUEUE, purges the OFS.RESPONSE.QUEUE file according to the minutes entered into the ATTRIBUTE.VALUE field on the TSA.SERVICE record. If a record is older than the time entered it will be deleted.

TSA Service Definition

Ensure that the OFS.MESSAGE.SERVICE is started by committing the *TSA.SERVICE* record with the SERVICE.CONTROL set to AUTO. The TSM service must also be running.

The `ATTRIBUTE.TYPE` and `ATTRIBUTE.VALUE` field have no validation on them as they are free form fields to be used for any and all *TSA.SERVICE* records to contain any value that a service may require. For the OFS TSA service minutes are required in the `ATTRIBUTE.VALUE` field. If a numeric value is not entered into this field records will not be purged off the *OFS.RESPONSE.QUEUE* file when the same TSA service is run.



TSA.SERVICE record OFS.MESSAGE.SERVICE set to auto

Appendices

Appendix 1 – OFS Queue Manager

Offline / Store and Forward Capability

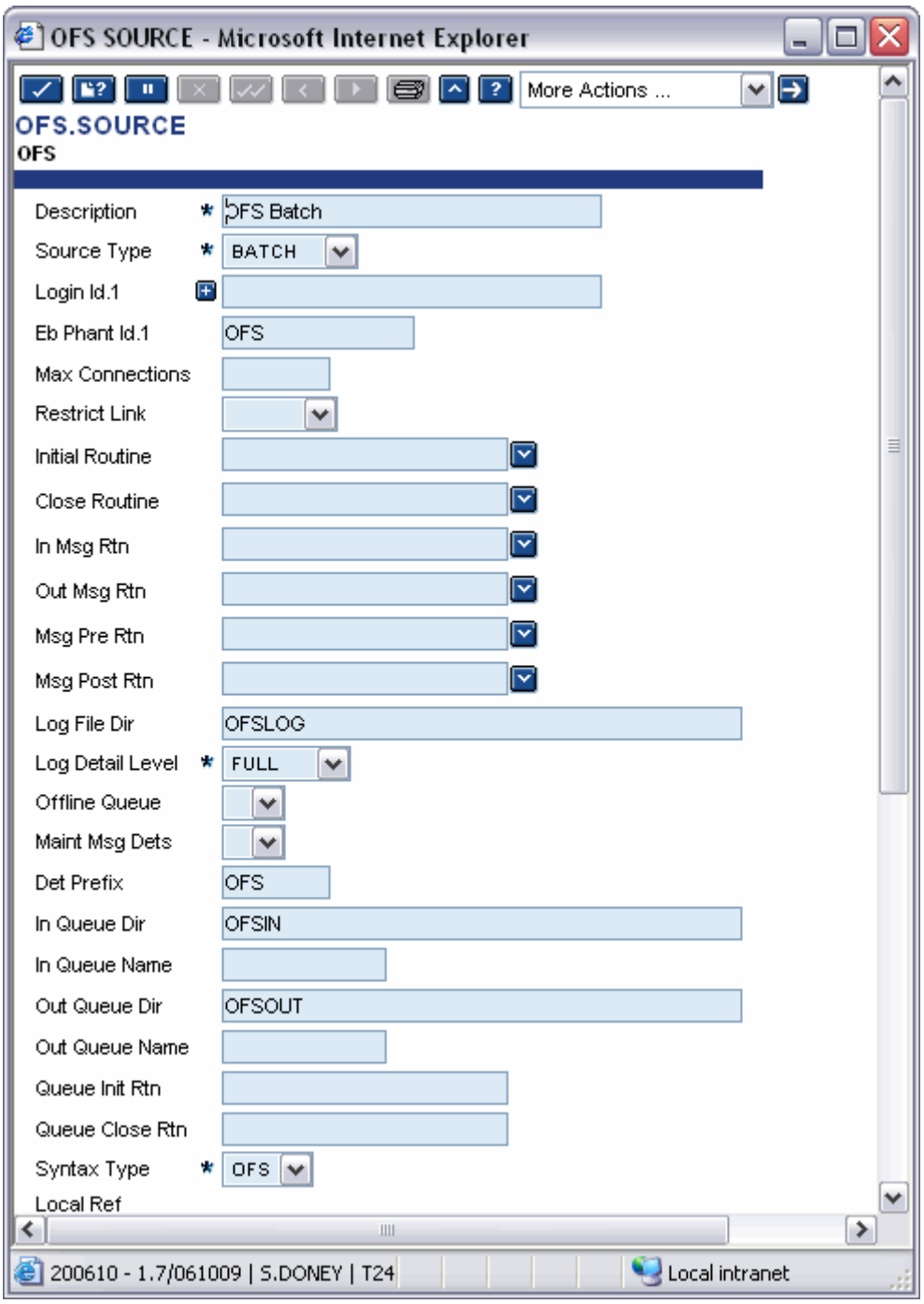
If the OFS.ONLINE.MANAGER is invoked from an external application and T24 is off-line at the time, an option in the *OFS.SOURCE* will determine if processing is allowed.

If processing is allowed the system will return details from Enquiry requests. For application update requests the system will allow messages to be added to a store and forward queue. In this situation the message will be recorded as passed to the OFS.ONLINE.MANAGER, a success / failure indicator will show that the message has been queued.

A background process, the OFS.QUEUE.MANAGER will be initiated when the system returns to its online state and will submit each queued message back to the OFS.REQUEST.MANAGER, results are written to an outward queue.

Batch communication with OFS

Batch and offline processing is performed by OFS.QUEUE.MANAGER running as a phantom. The mode of operation is determined by the relevant *OFS.SOURCE* record configuration. Processing is repeated at predetermined intervals defined in the relevant *EB.PHANTOM* record.



OFS Source Record Configuration

When running in batch mode a predetermined input directory defined in IN.QUEUE.FILE is selected for batch input files, which will contain the OFS messages.

If IN.FILE.NAME is defined then only that file will be processed from the input directory.

Each input file is processed as follows:

- Individual messages are extracted
- A pre processor routine is called if required
- The message is passed to OFS.REQUEST.MANAGER to update T24
- A post update routine is called if required
- The output file is updated as follows:
  - the returned message for OFS format
  - the returned message tagged for XML format
  - the input message appended with a return code
- When all messages have been processed the output file is written to the output directory defined in OFS.SOURCE

When running in offline mode the following processing takes place.

- OFS.OFFLINE.QUEUE is selected for the relevant OFS.SOURCE.
- Each record contains the identity of an OFS.REQUEST.DETAIL record, which is read and the OFS message is extracted.
- The message is processed as above.
- The OFS.OFFLINE.QUEUE record is deleted and the OFS.REQUEST.DETAIL is updated with the returned message.

Appendix 2 – OFS.SOURCE fields

Field Name	Field Purpose	Requirements for BATCH OFS
@ID	Id key of OFS.SOURCE record	ID allows alphanumeric characters. User may define own appropriate name.
DESCRIPTION	Simple description field.	Use an appropriate description to identify the purpose of this OFS.SOURCE record.
SOURCE.TYPE	Defines whether you will be using this OFS.SURCE record for BATCH or TELNET processing.	F or BATCH: select BATCH
LOGIN.ID	Specifies the Unix or NT login that will automatically initiate the communication with the external source.	For BATCH: leave blank.
EB.PHANT.ID	Specifies the EB.PHANTOM record to be used for processing batch jobs.	For BATCH processing, select the EB.PHANTOM record to be used.
MAX.CONNECTIONS	Specifies the maximum number of OFS connections for the specified service, which can be active at any one time.	Number must be less than the maximum number of users defined in the SPF record,
LOG.FILE.DIR	Defines the name of the directory used to store the log files created when running OFS.	Either refer to an existing pre-defined directory, otherwise the directory will be created at authorisation.
LOG.DETAIL.LEVEL	Specifies the type of OFS message logging to be performed by OFS.	Can select as much or as little logging as required – does not affect OFS processing.
IN.QUEUE.DIR	For batch processing, the directory where batches will be placed for processing.	OFS Batch processing: Enter name of directory to be used for batch processing.
SYNTAX.TYPE	Specifies the syntax type of the messages.	For Batch processing may be either GTS or OFS.
GENERIC.USER	The generic user is the T24 user that will be used to access the T24 server. For external users it determines the SMS options. Generic user must have the authority to use EB.PHANTOM.PH	Must be a valid record on the USER file and have the authority to use EB.PHANTOM.PH
IB.USER.CHECK	Specifies whether transactions supplied to T24 using this service require the sign on name and password to be validated for the external user.	For batch processing leave null.