Integrating Text Management System/38 (5769-DBI IBM S/38 Utilities for AS/400)

Overview

This gives an example with suggestions and recommendations for using Text Management System/38 as an alternative to OfficeVision/400.

You can start this program from the command line with the commands CRTDOC, EDTDOC, DSPDOC and PRTDOC.

"Work with Documents in Folders" display:

- Create (option 1)
- Revise (option 2)
- View (option 5)
- Print (option 6)

All other office requests, for example MRGDOC and other mail requests, either cannot be perfomed or can be performed using the SNDDST command. Additional functions can be implemented by modifying the program example.

Implementation

This program example was implemented for using TMS/38 as editor in the following panels.

It is important to keep in mind that TMS/38 does not use folders to store its documents, but uses members in a source physical file. Therefore it is necessary to have a routine in order to convert documents in folders to members in a file and vice versa.

The following steps were used to integrate TMS/38 with the commands to implement the necessary document handling capabilities:

- Customize the conversion routines
- Register the document handling exit program
- Create a program to start the application (TMS/38 in this case)
- Grant the proper authorities
- Test and document the program

Conversion Routines

In our example we used the conversion routines explained in the QUSRTOOL library. The QUSRTOOL library is available for a charge. The installation procedure of this tool must be completed before you can use the conversion commands. The documentation member AAAAREADME in the source file QUSRTOOL/QATTINFO provides instructions on how to install the TAA tool functions using CRTTAATOOL. QATTRPG, QATTCBL & QATTCL are actually save files and they must be restored into a library.

The installation program must be created into a library. As in all our examples we created all objects in the library OVENABLE, a library that you must create: CRTLIB OVENABLE.

```
CRTCLPGM PGM(OVENABLE/TOFCRTCL) SRCFILE(QUSRTOOL/QATTCL)
```

When this program is executed it creates commands and other objects from QUSRTOOL that are necessary for the conversion of documents into source files and vice versa. The execution requires one parameter to be passed to the program: the library name

```
CALL PGM(TOFCRTCL) PARM(OVENABLE)
```

Note: Be aware that this tool might lose formatting information when convertion is performed either way.

Register the Document Handling Exit Program

Getting it Done

This section intends to help the user get ready for installing and writing his own document handling exit programs. It first shows how to register the exit program. In a second step the reader is shown how the exit program is called and how the program is structured.

Registering the Document Handling Exit Program

To register document handling exit and conversion programs two APIs are used: the first one to actually register programs, the second one to view the registered programs. The use of both APIs will be discussed in the following sections.

Change Office Exit Program (QOGCHGOE) API

To register a document handling exit program the Change Office Exit Program (QOGCHGOE) API is used. This API registers the exit program to be called for document handling requests as well as the exit program for document conversion requests. When it is called the QOGCHGOE API writes the parameters passed to it in a data area in library QUSRSYS. The data area is created the first time the Change Office Exit Program API is called. Do not write directly into the data area as IBM might change the access method in the future. Use the fully supported QOGCHGOE API instead.

The Change Office Exit Program (QOGCHGOE) API is called with four parameters, as listed below in Table 2.

Table 2. Required Parameters for QOGCHGOE API

1	Document handling exit program and library name	Input	Char(20)	The first 10 characters contain the program name or *IBM, and the second 10 the library where the program is located.
2	Document handling program supports mail flag	Input	Char(l)	If an application is not intended to handle mail, set this flag to 0, otherwise to 1. If set to 1 the program specified in the document handling exit program and library name parameter is called for mail requests.
3	Document conversion exit program and library name	Input	Char(20)	The first 10 characters contain the program name or *IBM, and the second 10 the library where the program is located.
4	Error code	I/O	Char(*)	Parameter in which the API returns error information for further processing.

Further information on the Change Office Exit Program API and its parameters can be found in Chapter 21 of the *AS/400 System Programmer's Interface Reference.* Registering a document handling exit program can be as easy as shown in the CL program below. This program sets the document handling exit to program OVTMS38 in library OVENABLE. The mail flag parameter is set to 0, which means that no mail requests will be handled by the program OVTMS38. The document conversion exit program parameter is set to its default *IBM. Thus no user-written document conversion program will take control if document conversions are needed.

```
*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+...

 PGM
       DCL  VAR(&EDITOR) TYPE(*CHAR) LEN(20) +
              VALUE(OVTMS38    OVENABLE  ')
       DCL VAR(&MFLAG) TYPE(*CHAR) LEN(1) VALUE('0')

       DCL  VAR(&CONVER) TYPE(*CHAR) LEN(20) +
             VALUE(*IBM    )
       DCL  VAR(&ERROR) TYPE(*CHAR) LEN(100) +
             VALUE(X'00000074')

      CALL  PGM(QOGCHGOE) PARM(&EDITOR &MFLAG &CONVER +
             &ERROR)
 ENDPGM
```

Explanations:

[] Setting the document handling exit program to **OVENABLE/OVTMS38**.

[] Setting the document conversion program to *IBM. Thus only IBM programs and programs registered in the document conversion table are used to convert documents.

[] The variable to receive any error information from the QOGCHGOE API. Although the variable is declared, this program does not do any error handling. Refer to Appendix A, "Document Handling Exit Registering Tools" on page 62 for more information about error handling in this context.

Note: Remember to initialize the first 4 bytes of the error field with its length in binary before calling QOGCHGOE. In the example above X'74' equals the decimal 100. If the first 4 bytes equal X'00' any error indication will be returned as an escape message.

Writing the Document Handling Exit Program

After the document handling exit program is registered using one of the methods explained above, the next step is to write it. This section gives general guidelines for writing document handling exit programs. Detailed explanations are to be found in the TMS/38 example.

If a document handling exit program is registered and a user makes a document handling request, either using a CL command such as EDTDOC or typing in an option such as option 2 on the "Work with Documents in Folders" screen, the request is handed to the document handling exit program with six parameters that are listed in Table 4.

Table 4. Required Parameters for Document Handling Exit Programs

1	Document name	input	Char(12)	The name of the document the required function is performed against.
2	Folder name	Input	Char(63)	The folder in which the document is located.
3	Document type	Input	Binary(4)	The format of the document that is being worked with. The value must be from 1 to 65535. Refer to Table 1 on page 9 for more information.
4	Function	Input	Char(10)	The type of operation that the user is requesting for this document. Valid functions **are:** CREATE, VIE-W, EDIT, MAILVIEW, MAILEDIT, MAILFWD, MAILREPLY, MERGE, MERGEOPTS, PRINT, PRINTOPTS, PAGINATE, SPELLCHECK.
5	Function-specific information	Input	Char(485)	Additional information depending on the function requested in parameter 4.
6	Exit processing indicator	Output	Char(4)	The additional processing that OV/400 should perform on return of the document handling exit program. Valid indicators are: 0000, 0001, 0002, 0007, 0010.

The first three parameters, document name, folder name, and document type, are self explanatory and will not be further discussed here. Parameter 5 (function-specific information) differs depending on the type of operation described in parameter 4.

Detailed information on all parameters can be found in Chapter 22 of the *AS/400 System Programmer's Interface Reference.* In this manual you will find the layout of the following formats:

DOCl0100 Format for print function requests
DOCl0200 Format for merge function requests
DOCl0300 Format for spell function requests
DOCI0400 Format for mail function requests
DOCI0500 Format for edit function requests
DOCI0600 Format for create function requests

Note: Library QUSRTOOL includes source members describing the layout of the function-specific information for using with RPG/400 or COBOL/400 programming. In order to be used in customer written programs they should be copied out of file QATTRPG, for RPG, and QATTCBL, for COBOL, into the application library. The names of the members are OPDOCI0100 to OPDOCIOB00 describing the formats DOCI0100 to DOCI0600 respectively.

Writing a Document Conversion Program

A conversion program can be registered as a document conversion exit program using the QOGCHGOE APl as explained in "Change Office Exit Program (QOGCHGOE) APl". In this case the registered document conversion program takes control over all document conversions.

Table 8. Required Parameters for Document Conversion Programs

1	Input document name	input	Char(12)	The name of the document the required function is performed against.
2	Input folder name	Input	Char(63)	The folder in which the document is located.
3	Input document type	Input	Binary(4)	The format of the document that is being worked with. The value must be from 1 to 65535. Refer to Table 1 on page 9 for more information.
4	Output document name	Input	Char(12)	The name of the output document.
5	Output folder name	Input	Char(63)	The folder in which the document is located.
6	Output document type	Input	Binary(4)	The format of the document that is being worked with. The value must be from 1 to 65535. Refer to Table 1 on page 9 for more information.
7	Function code	Input	Char(l)	Indicates whether the exit is being called to check for the existence of a conversion or to perform the conversion. The codes are: 0 (a conversion is being requested) and 1 (an existence check is requested).
8	Conversion existence indicator	Output	Char(l)	Be aware that this is the only output parameter. It indicates whether the requested conversion exists or not. It has to be set for both conversion existence checks and conversion requests. The valid values are: 0 (conversion does not exist) and 1 (conversion exists).

Create the Document Handling Exit Program

In order to use TMS/38 it is necessary to write a document handling exit program which is called by the xxxDOC commands. This exit program receives the parameters passed, performs the conversion from documents in folders and then calls TMS/38 to process the requested function (Edit, View, etc.). Once a document is stored with TMS/38, the program copies the text from the source physical file to its original document in a folder and returns the control to the command entry.

The explanations corresponding to each one of these steps are to be found after the listing of the program.

```
  .....................          SEU SOURCE LISTING
S0URCE FILE  OVENABLE/QCLSRC
MEMBER       OVTMS38
*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
   /* DOCUMENT HANDLING EXIT PROGRAM TMS38 Example            */

    PGM         PARM(&DOC &FLR &DOCTYP &FUNC &FNCPRMS &EXITIND)

 /*   PARMS. PASSED BY xxxDOC COMMAND                         */
      DCL       VAR(&DOC) TYPE(*CHAR) LEN(12)
      DCL       VAR(&FLR) TYPE(*CHAR) LEN(63)
      DCL       VAR(&DOCTYP) TYPE(*CHAR) LEN(4)
      DCL       VAR(&FUNC) TYPE(*CHAR) LEN(18)
      DCL       VAR(&FNCPRMS) TYPE(*CHAR) LEN(485)
      DCL       VAR(&EXITIND) TYPE(*CHAR) LEN(4)

  /*  LOCAL VARIABLES                                       */
      DCL VAR(&USER) TYPE(*CHAR) LEN(10)
      DCL VAR(&OPTION) TYPE(*CHAR) LEN(1)
      DCL VAR(&MEMEXI) TYPE(*LGL) LEN(1) VALUE('1')
      DCL VAR(&DLONAM) TYPE(*CHAR) LEN(10)
      DCL VAR(&NBR) TYPE(*CHAR) LEN(6)
      DCL VAR(&RECURS) TYPE(*CHAR) LEN(l0)
      DCL VAR(&LOOP) TYPE(*DEC) LEN(4 0)
      DCL VAR(&DOCTYD) TYPE(*DEC) LEN(5)



    /*   START OF PROGRAM           */
    /*  CHECK DOCUMENT TYPE         */

        CHGVAR VAR(&DOCTYD) VALUE(%BIN(&DOCTYP))
        IF     COND(&DOCTYD *EQ 2) THEN(GOTO CMDLBL(CONT1))
        IF     COND(&DOCTYD *EQ 11) THEN(GOTO CMDLBL(CONT1))
        IF     COND(&DOCTYD *EQ 32768) THEN(GOTO CMDLBL(CONT1))
        IF     COND(&DOCTYD *EQ 32769) THEN(GOTO CMDLBL(CONT1))
        ELSE   CMD(DO)
         CHGVAR VAR(&EXITIND) VALUE('0010')
         GOTO   CMDLBL(ENDREC)
        ENDDO
CONT1:
      CHKOBJ OBJ(OVENABLE/OVTMS38) OBJTYPE(*FILE)
      MONMSG MSGID(CPF9881) EXEC(DO)

       CRTSRCPF FILE(OVENABLE/OVTMS38)

      ENDDO



      RTVJOBA USER(&USER) NBR(&NBR)
      CHGVAR  VAR(&RECURS) VALUE('OVRC' *CAT &NBR)
      CHKOBJ  OBJ(QTEMP/&RECURS) OBJTYPE(*DTAARA)
      MONMSG  MSGID(CPF9801) EXEC(DO)
   CRTDTAARA  DTAARA(QTEMP/&RECURS) TYPE(*DEC) LEN(4)
  ENDDO
RTVDTAARA DTAARA(QTEMP/&RECURS) RTNVAR(&LOOP)
  CHGVAR    VAR(&LOOP) VALUE(&LOOP + 1)
  CHGDTAARA DTAARA(&RECURS) VALUE(&LOOP)
IF          COND((&FUNC *EQ 'PRINT') *AND (&LOOP > 1) +
           *OR (&FUNC *EQ 'MERGE') *AND (&LOOP > 1)) +
                    THEN(DO)
  CHGVAR VAR(&EXITIND) VALUE('0000')
  GOTO     CMDLBL(ENDREC)
  ENDDO

  IF         COND((&FUNC *EQ 'MERGE') *OR (&FUNC *EQ +
             'MERGEOPTS') *OR (&FUNC *EQ 'PRINTOPTS') +
             *OR (&FUNC *EQ 'PAGINATE') *OR (&FUNC *EQ +
             'SPELLCHECK')) THEN(DO)
  CNGVAR VAR(&EXITIND) VALUE('0000')
  GOTO     CMDLBL(END)
  ENDDO

  IF         COND(&USER *NE 'ITSCID16') THEN(DO) /* This +
                                          is the TMS/38 user */

  CHGVAR VAR(&EXITIND) VALUE('0000')
  GOTO CMDLBL(ENDREC)
  ENDDO

  RTVDLONAM DLO(&DOC) FLR(&FLR) RTNOBONAM(&DLONAM) /* +
                       Create unique member name in OVTMS38 */

   CHKOBJ OBJ(OVENABLE/OVTMS38) OBJTYPE(*FILE) MBR(&DLONAM) +
                 /* Check if member already exist, i.e. +
                    ' document is in use */
   MONMSG MSGID(CPF9815) EXEC(CHGVAR VAR(&MEMEXI) +
     VALUE('0'))
  IF      COND(&MEMEXI *EQ '1') THEN(DO)
    SNDPGMMSG MSGID(xxxxxxx) MSGF(xxxxxxx) MSGDTA(&DOC) +
        TOPGMQ(*PRV) MSGTYPE(*DIAG)
    SNDPGMMSG MSGID(xxxxxxx) MSGF(xxxxxxx) TOPGMQ(*PRV) +
        MSGTYPE(*ESCAPE)
     GOTO CMDLBL(END)
   ENDDO

  IF     COND(&FUNC *EQ 'CREATE') THEN(DO)
              CHGVAR VAR(&OPTION) VALUE('1')
        ENDDO
      ELSE DO
         IF      COND(&FUNC *EQ 'EDIT') THEN(DO)
                 CHGVAR VAR(&OPTION) VALUE('1')                    ENDDO
        ELSE DO
          IF COND(&FUNC *EQ 'PRINT') THEN(DO)
            CHGVAR VAR(&OPTION) VALUE('3')

            ENDDO
          ELSE DO
          IF COND(&FUNC *EQ 'VIEW') THEN(DO)
           CHGVAR VAR(&OPTION) VALUE('5')
           ENDDO
           ELSE DO

               CHGVAR VAR(&EXITIND) VALUE('0010')
               GOTO END
```

```
                ENDDO
          ENDDO
      ENDDO


          TOFCVTSF FROMDOC(&DOC) FROMFLR(&FLR) +
                TOFILE(OVENABLE/OVTMS38) TOMBR(&DLONAM) REPLACE(*NO)
           MONMSG   MSGID(CPF9999)
           ENDDO

/* INVOKE TMS38                                                       */
      QSYS38/EDTTXT SRCFILE(OVENABLE/OVTMS38) +
                           DOCUMENT(&DLONAM) OPTION(&OPTION) ALWCL(*YES)
      IF         COND(&OPTION *EQ '1') +
        THEN(DO)
         TOFCVTDC FROMFILE(OVENABLE/OVTMS38) FROMMBR(&DLONAM) +
          TODOC(&DOC) TOFLR(&FLR) REPLACE(*YES)
           MONMSG MSGID(CPF0000)
   ENDDO

 RMVM FILE(OVENABLE/OVTMS38) MBR(&DLONAM) /* +
          Remove member. */
                    MONMSG MSGID(CPF7310) EXEC(DO)
                    DLTDLO DLO(&DOC) FLR(&FLR)
                    SNDPGMMSG MSGID(CPC8A15) MSGF(QCPFMSG) TOPGMQ(*PRV) +
                               MSGTYPE(*DIAG)
                    SNDPGMMSG MSGID(OFC1680) MSGF(QOFCMSG) TOPGMQ(*PRV) +
                               MSGTYPE(*ESCAPE)

                     ENDDO
 END: CNGVAR VAR(&EXITIND) VALUE('0000')

   DLTDTAARA DTAARA(QTEMP/&RECURS)

 ENDREC:     ENDPGM
                       **** END OF SOURCE ****
```

OV/400 calls this program and passes the corresponding parameters.

The local variables are the following:

&user = User profile. Profile name of the user requesting the OV/400 function.
&option -- Option. The option for TMS/38 (1: Create, 3: Print, 5: View in a print format).
&memexi -- Member existence. This logical variable is used when the member exists in the physical file.
&dionam = Document library object name. It is the system object name of the document.
&nbr = Job number. It is used to identify the job's data area.
&recurs = Data area containing the counter set by the &loop variable. &loop = Number of times the document handling exit program was called.

- TMS/38 can process documents with types FFTDCA (2), RFTDCA (11), RFTAS400 (32768) and FFTDCA (32769). Documents with other formats are ingored.

- The source physical file (OVTMS38) is used in this example to temporarily copy a document in a folder into a member in this file. The name of the member will be the document's system object name. After the processing (Edit, Print, etc.) by TMS/38, the member is copied back to the same document in the same folder and removed from the source file.

- The first program step is to verify the source physical file existence. If it does not exist it is created at run time.

- The number of the job is retrieved to form a unique name for the data area. If the data area does not exist it is created in QTEMP.

- The loop counter is retrieved from the data area and increased by one.
- The routines for converting source files into documents and vice versa use the PRTDOC and MRGDOC commands. As the document handling exit program calls the conversion routines and the conversion routines use the MRGDOC and PRTDOC commands, which the OV/400 Application Enabler hands over to the document handling exit program, the program ultimately calls itself. To avoid false processing in the recurrent calls of the document handling exit program, the PRTDOC and MRGDOC request is effectively no-op'd by setting the exit indicator to 0000.

- This allows the user to make a PRTDOC request because the "PRINT" function is processed the first time the program is called, but not the subsequent times.

- This step changes the &exitind to "0000" if the requested function cannot be handled by TMS/38.

- This step gives the alternative to restrict the use of TMS/38 to certain users. In this example only user ITSCID16 will invoke TMS/38. For all other users the operation is ignored by setting the value "0000" in the &exitind.

- Each member copied to the source physical file is created with a unique name. This step retrieves the system object name and with this unique name the member is created.

- If the document to be copied already exists in the source file, the program assumes that somebody is working with the document. Then the program sends the message "Document is in use" and returns . You may have to create your own message .

- These instructions are meant to change the &option variable value to the corresponding &func parameter value. This allows for invoking TMS/38 directly with the desired function.

- If the function to execute is not "Create", the command TOFCVTSF copies the document to process to the source file in the library OVENABLE.

- The command EDTTXT is executed. It calls TMS/38 to process the document. The passed parameters are the document name and the option that TMS/38 is going to execute. After the document is processed if the user request was "Create" or "Edit" the member is copied to the document. If the option was "Print" or "View" the document is not converted back to its folder.

- The member is erased from the source file. If the member is empty it can not be copied to a document.

- Before control is given over, the data area for counting the invocation level is deleted.

Operation and Restrictions

The library OVENABLE must be in the library list.

Be sure that users do not cancel their jobs while using TMS/38. When a user makes a document request, the program checks if a member with the document's system object name exists in the source file OVTMS38. If this is not the case, the member is created. If it already exists, the program assumes that the document is in use and notifies the user of this fact. When a user cancels his job the member will remain in the source file OVTMS38, so other users will be notified that the document is in use.

If a user has requested to end the current editing session, the document will still be locked. In this case it is necessary to delete the member in the source file OVTMS38.

If a document does not contain any information it will be erased. This is necessary because the conversion tools can not process empty documents.

Retrieve Office Programs (QOGRTVOE) Apl

To view the registered document handling and conversion programs the Retrieve Office Programs (QOGRTVOE) APl is used. It is called with four parameters that are listed in Table 3.

Table 3. Required Parameters for QOGRTVOE APl

1	Receiver variable	Output	Char(*)	This variable receives all exit program information. To accommodate the information received in V2R2 its length has to be at least 49 characters.
2	Length of receiver variable	Input	Binary(4)	For V2R2 length has to be set to at least 49. The value has to be handed in binary form.
3	Format name	Input	Char(8)	The format in which the information will be returned into the receiver variable. In V2R2 the only valid format name is *OGOE0100.*
4	Error code	I/O	Char{*)	Parameter in which the APl returns error information for further processing.

As all information of the Retrieve Office Programs APl is given back into one receiver variable, the latter has to be processed further. All details about the OGOE0100 format in which the information is written into the receiver variable, can be found in Chapter 21 of the *AS/400 System Programmer's Interface Reference.* How to process the information received by the Retrieve Office Programs APl is shown in the example in Appendix A, "Document Handling Exit Registering Tools" on page 62.

```
          *...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+...
           PGM       /* VARIABLES TO PASS TO THE QOGRTVOE API
                 DCL   VAR(&RECVAR) TYPE(*CHAR) LEN(100)
                 DCL   VAR(&LENGTH) TYPE(*CHAR) LEN(4)
                 DCL  VAR(&FORMAT) TYPE(*CHAR) LEN(8) +
                           VALUE(OGOE0100)
                 DCL VAR(&ERROR) TYPE(*CHAR) LEN(100)
                        /* INTERNAL VARIABLES
                   DCL         VAR(&DECLEN) TYPE(*DEC) LEN(9 0) VALUE(100)
                 DCL VAR(&EXIPGM) TYPE(*CHAR) LEN(10) /* Document
                                            Handling Exit Program */
                 DCL VAR(&EXILIB) TYPE(*CHAR) LEN(10) /* Library +
                                            for Exit Program */
                 DCL VAR(&CVNPGM) TYPE(*CHAR) LEN(10) /* Document
                                            Conversion Exit Program */
                DCL VAR(&CVNLIB) TYPE(*CHAR) LEN(10) /* Library +
                                            for Conversion Program */
DCL VAR(&MAILFL) TYPE(*CHAR) LEN(1) /* Mail Flag */
                 DCLF  FILE(OVENABLE/OVRTVOE)

                CHGCURLIB CURLIB(OVENABLE)
                  CHGVAR VAR(%BIN(&LENGTH)) VALUE(&DECLEN)
                CALL  PGM(QOGRTVOE) PARM(&RECVAR &LENGTH &FORMAT +
                       &ERROR)
                CNGVAR VAR(&EXIPGM) VALUE(%SST(&RECVAR 9 10))
                 CHGVAR VAR(&EXILIB) VALUE(%SST(&RECVAR l9 10))
                 CHGVAR VAR(&MAILFL) VALUE(%SST(&RECVAR 29 1))
                 CHGVAR VAR(&CVNPGM) VALUE(%SST(&RECVAR 30 10))
                 CHGVAR VAR(&CVNLIB) VALUE(%SST(&RECVAR 40 10))
```

```
  SNDRCVF DEV(*FILE) /* Display Current Exit Program */
ENDPGM
```

Explanations:

[] Receiver variable for all information. It has to be at least 49 characters long.

[] Length of the receiver variable. Although its format is binary for the API, it has to be declared character in the CL program.

[]Only possible format name in V2R2 is OGOE0100.

[] The following variables are used for transforming the information in the receiver variable to be able to display them in a display file.

[] Transforming the type of the receiver variable from decimal to binary. The %BIN function allows the transformation in both ways, from decimal to binary and vice versa.

[] The Retrieve Office Programs API is called and the information in the receiver variable handed over to specific variables that are shown in the display file.