

COS

Create Objects Support

Programmer Guide

PG-COS

21st April 2009

Chapter A.

COS – Concept and Functions

The **COS** tool has been created by GEFIS in order to support their own software development.

It's a tool for programmer and is used

- to simplify compiling and bind operations by standardized parameter in create commands.
- to avoid errors caused by wrong or incomplete create parameter definitions and/or library lists.
- to support (optionally) change documentation by automatic log entries in source code and a separate log file.

COS supports creation of following objects:

- **ILE-RPG Programs**
- **ILE-RPG Service Programs**
- **ILE-CL Programs**
- **Display Files**
- **Printer Files**

Core function in COS is the adoption of the command **CRTOBJCOS**, which is to use instead of the appropriate OS commands for creation of objects out of source files.

All for COS necessary key information will be supplied by a pseudo code in the source data and is interpreted by the create command.

Object creation is optionally documented at the end of the source file as *Source History Entry*.

COS was developed with OS Version **5**, Release **4** and requires this level or higher.

Chapter B. Creation of Objects with COS

1. The CRTOBJCOS command (Create Object using COS)

For creation of **ILE Program Objects** and **File Objects** from type **DSPF** and **PRTF** there should be used exclusively the **CRTOBJCOS** command.

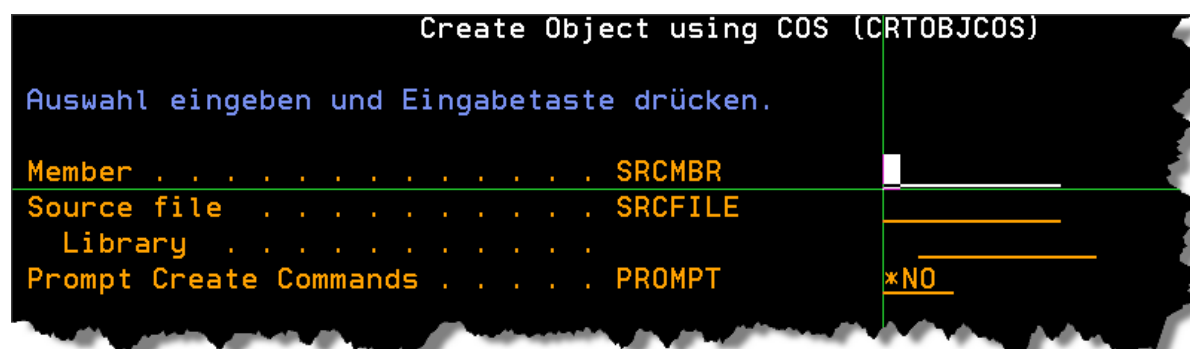
Execution of this command may be interactive or in batch mode.

Command parameter **PROMPT** may be used in interactive mode for prompting the original OS create commands (e.g. CRTRPGMOD, CRTPGM, etc.).

This is for exceptional cases, if special create options necessary, which are not supported by COS.

If command CRTOBJCOS is executed in batch mode, command parameter PROMPT is ignored.

Invoke of the CRTOBJCOS command is normally done by user defined PDM option in the PDM menu "Work with Members Using PDM".



The screenshot shows a terminal window titled "Create Object using COS (CRTOBJCOS)". Below the title is a prompt "Auswahl eingeben und Eingabetaste drücken." followed by a form. The form has four rows, each with a label on the left and a field on the right. The first row is "Member" with field "SRCMBR". The second row is "Source file" with field "SRCFILE". The third row is "Library" with field "SRCFILE". The fourth row is "Prompt Create Commands" with field "PROMPT". To the right of the "PROMPT" field, there is a small box containing the text "*NO".

Field	Value
Member	SRCMBR
Source file	SRCFILE
Library	SRCFILE
Prompt Create Commands	PROMPT

After calling this command there are initially some plausibility checks. If errors are detected appropriate messages will be sent to the user.

Certain create information (Source History Entries) will be added at the end of the source code as comments after the object has been created successfully if ***YES** was specified in COS keyword parameter **WRTSRCHSTE**.

They contain also the full OS command which was used for object creation.

The syntax of this documentaion data is described detailed in **Chapter D** below.

A log entry is written to file **CRTOBJLOG** if COS keyword parameter **WRTCRTLOGE** was specified with ***YES** and the object creation has been ended successfully.

File CRTOBJLOG must be found along the current job's library list. The log file record is described in **Chapter E** below.

2. COS control statements in source codes

For using COS command CRTOBJCOS you must enter before certain controlling data in the source code. These entries are called in the following as COS statements.

These COS statements will be checked on existence, syntax and plausibility when the COS command CRTOBJCOS is executed.

The object creation will only be performed if the COS statements are complete and error-free.

2.1. General rules on COS statements in source files

2.1.1. Start and end statements

COS statements must be embedded by following start and end lines which have to be located as comments **before the first real code** line. Each source file may contain only one COS group.

```
/$B    = Start of COS statements
...    COS statement 1;
...    COS statement 2;
...    COS statement n;
/$E    = End of COS statements
```

2.1.2. General COS syntax

Each COS statement (incl. start and end statements) can be described in source codes **only as comment** lines.

Due to this fact, COS statements in **RPG** and **DDS** sources may start from position **8** and **CLP** sources from position **3** of each line.

COS statements start generally with a COS keyword trailing by one or a list of keyword values which are embedded in parentheses.

Keyword values are always assumed as character strings. Therefore they should not be quoted.

A list of keyword values ends by a right parentheses.

A COS statement is terminated always by a ; (semicolon) except /\$B and /\$E lines.

The COS statements (between start and end statement) may be in any order.

A list of keyword values may be continued in the following line from position 8 without specifying a continuation flag. The end of a COS statement is determined by a ; (semicolon) only.

Syntax example in RPG source code:

```
* /$B
* Keyword1(value1);
* Keyword2(Value1 Value2);
* Keyword3(Value1 Value2 Value3
* Value4);
* /$E
```

The asterisk ***** on position **7** marks the line as comment line. Position **8** was left blank for optical reasons only.

Syntax example in CLP source code:

```
/* /$B */
/* Keyword1(Value1 Value2 Value3); */
/* Keyword2(Value1); */
/* Keyword3(Value1 Value2 Value3 Value4 */
/* Value5); */
/* /$E */
```

The character string **/*** in position **1** or higher marks the line as comment line. Comment lines are terminated by character string ***/**. Comment continuation lines may be specified by a plus **+** at the end of the line.

Example:

```
/* /$B */
/* Keyword(Value1 Value2 Value3 Value4 +
Value5); */
/* /$E */
```

2.2. COS statements in source code

COS-Keyword	Key Value(s)	M= Mandatory	Valid Types
TYPE	COS Object Type: RPG = ILE-RPG program from source type RPGLE or SQLRPGLE. RPGSRV = ILE-RPG service program from source type RPGLE or SQLRPGLE. CLP = ILE-CLP program from source type CLLE. DSPF = Display files PRTF = Printer files	M	ALL
OBJLIB	Name of object library for object types *PGM and *FILE	M	ALL
WRTSRCHSTE	Write source history entry: *YES or *NO (see Chapter D)		ALL
WRTCRTLOGE	Write create object log file CRTOBJLOG : *YES or *NO . File CRTOBJLOG must be found in the current library list. (see Chapter E)		ALL
LIBLARA	Data area with library list for object creation: Data area or Library/Data area Remarks: The library list in the data areas may be administrated by the command EDTLIBLDTA (see Chapter C).		ALL
ACTGRP	Like corresponding parameter from OS commands CRTPGM and CRTSRVPGM. For GEFIS restricted to Program name or QILE or *CALLER	M	RPG RPGSRV CLP
OPTION	Like corresponding parameter from OS commands CRTPGM and CRTSRVPGM, e.g. OPTION(*DUPPROC *DUPVAR)		RPG RPGSRV CLP
USRPRF	Like corresponding parameter from OS commands CRTPGM and CRTSRVPGM, e.g. USRPRF(*OWNER)		RPG RPGSRV CLP
BNSRVPGM	Like corresponding parameter from OS command CRTPGM. (In RPG programs we use service programs principally by binding directories, specified in the H-Specifications of the source code)		CLP
BNDDIR	Like corresponding parameter from OS command CRTPGM. (In RPG programs we specify binding directories principally in the H-Specifications of the source)		CLP
EXPORT	Like corresponding parameter from OS command CRTSRVPGM, e.g. EXPORT(*SRCFILE) If *SRCFILE is specified, name of the export source file is assumed as QSRVSRC .	M	RPGSRV

COS-Keyword	Key Value(s)	M= Mandatory	Valid Types
OVRDBF	CL command for override a data base file before creation of module (without parameter OVRSCOPE). Example: OVRDBF FILE(FILEX) TOFILE(LIBY/FILEY) Delete override is done automatically after object creation.		RPG RPGSRV CLP
LVLCHK	Record format level check (*NO, *YES). Example: LVLCHK(*NO) (Default is *NO)		DSPF PRTF
DEVTYPE	Like corresponding parameter from OS command CRTPRTF, e.g.: DEVTYPE(*IPDS)		PRTF
PAGESIZE	Like corresponding parameter from OS command CRTPRTF, e.g.: PAGESIZE(72 132) (Default is 72 132) *ROWCOL is always assumed as third keyword parameter.		PRTF
OVRFLW	Like corresponding parameter from OS command CRTPRTF, e.g.: OVRFLW(66) (Default is 66)		PRTF
HOLD	Like corresponding parameter from OS command CRTPRTF, e.g.: HOLD(*YES) (Default is *YES)		PRTF

2.3. Code examples for COS statements

RPG Program

```
0001.00 *****
0002.00 * GEFIS GESELLSCHAFT FÜR INDIVIDUAL-SOFTWARE MBH *
0003.00 *****
0004.00 * Program : EXAMPLE1 *
0005.00 * Function: *
0006.00 *****
0007.00 */$B
0008.00 * TYPE(RPG);
0009.00 * OBJLIB(GEFISWN);
0010.00 * WRTSRCHSTE(*YES);
0011.00 * WRTCRTLOGE(*YES);
0012.00 * LIBLARA(GEFIS);
0013.00 * ACTGRP(QILE);
0014.00 * OPTION(*DUPPROC *DUPVAR);
0015.00 * USRPRF(*OWNER);
0016.00 * OVRDBF FILE(GPKP00) TOFILE(ZL01W511/GPKP00);
0017.00 */$E
0018.00 H Copyright('(C) GEFIS GmbH 2009')
0019.00 H BndDir('GSL':'GASS':'QC2LE')
0020.00 H Debug Option(*NODEBUGIO: *SHOWCPY: *EXT: *SRCSTMT)
```

RPG Service Program

```
0001.00 *****
0002.00 * GEFIS GESELLSCHAFT FÜR INDIVIDUAL-SOFTWARE MBH *
0003.00 *****
0004.00 * Program : EXAMPLE2 *
0005.00 * Function: *
0006.00 *****
0007.00 */$B
0008.00 * TYPE(RPGSRV);
0009.00 * OBJLIB(GEFISWN);
0010.00 * WRTSRCHSTE(*YES);
0011.00 * WRTCRTLOGE(*YES);
0012.00 * LIBLARA(GEFIS);
0013.00 * ACTGRP(QILE);
0014.00 * OPTION(*DUPPROC *DUPVAR);
0015.00 * USRPRF(*OWNER);
0016.00 * EXPORT(*SRCFILE);
0017.00 * OVRDBF FILE(GPKP00) TOFILE(ZL01W511/GPKP00);
0018.00 */$E
0019.00 H Copyright('(C) GEFIS GmbH 2009')
0020.00 H BndDir('GSL':'GASS':'QC2LE')
0021.00 H Debug Option(*NODEBUGIO: *SHOWCPY: *EXT: *SRCSTMT)
```

CL Program

```
0001.00 /*****
0002.00 /* GEFIS GESELLSCHAFT FÜR INDIVIDUAL-SOFTWARE MBH */
0003.00 /*****
0004.00 /* Program : EXAMPLE3 */
0005.00 /* Function: */
0007.00 /*****
0008.00 /* */$B */
0009.00 /* TYPE(CLP); */
0010.00 /* OBJLIB(GEFISWN); */
0011.00 /* WRTSRCHSTE(*YES); */
0012.00 /* WRTCRTLOGE(*YES); */
0013.00 /* LIBLARA(GEFIS); */
0014.00 /* ACTGRP(QILE); */
0015.00 /* OPTION(*DUPPROC *DUPVAR); */
0016.00 /* USRPRF(*OWNER); */
0017.00 /* BNDSRVPGM(GSL/YYSCLS); */
0018.00 /* */$E */
0019.00 /*****/
```


Display File

```
*****
* GEFIS GESELLSCHAFT FÜR INDIVIDUAL-SOFTWARE MBH *
*****
* DSPF      : EXAMPLE4 *
* Function: *
*****
*/$B
* TYPE(DSPF);
* OBJLIB(GEFISWN);
* LVLCHK(*NO);
*/$E
*
```

Printer File

```
*****
* GEFIS GESELLSCHAFT FÜR INDIVIDUAL-SOFTWARE MBH *
*****
* PRTF      : EXAMPLE5 *
* Function: *
*****
*/$B
* TYPE(PRTF);
* OBJLIB(GEFISWN);
* LVLCHK(*NO);
* DEVTYPE(*IPDS);
* HOLD(*YES);
*/$E
*
```

Chapter C.

Administrate library lists

COS uses for creation of program objects normally a library list which is determined by a COS statement with keyword LIBLARA. The keyword value must be a data area (qualified or unqualified) which holds a valid library list.

The maximum length of such a data area is 2000 Bytes (= 181 libraries in list).

The content of these data areas (= library lists) may be edited by the command **EDTLIBLDTA**, which is also supplied by COS. This command corresponds largely to the OS command EDTLIBL.

Example:

```
EDTLIBLDTA      Edit LIBL in Data Area
Data area : GEFIS
Library   : TPSRCADM

Type new/changed information, press Enter
```

Sequence Number	Library	Sequence Number	Library	Sequence Number	Library
0		120		240	
10	QTEMP	130		250	
20	QGPL	140		260	
30	#GASS PE	150		270	
40	#SERVICE	160		280	
50	GASS	170		290	
60	GSL	180		300	
70	TRICS	190		310	
80	WDOG	200		320	
90		210		330	
100		220		340	
110		230		350	

F5=Refresh F3=Exit

Weitere ...

Remark: The COS command CRTOBJCOS checks - before creating an object - if the entries in the library list are (still) valid.

Chapter D.

Syntax of Source History Entries in Source Codes

COS supports optionally documentation on object creations by automatic updates in source file members as comment lines.

Comment lines are marked in

- **RPGLE** , **DSPF** and **PRTF** sources by an asterix (*) in position **7** and in
- **CLLE** sources by the character string **,/*** from position **1** and character string **,/*** at the end of the comment. The end of a comment line in CLLE sources could also be located in a continuation line, which is identified by a plus at the end of the previous line. But the comment lines in CLLE sources generated by COS are always terminated with character string **,/*** in position **79**.

All documentation entries are located at the end of the source lines or before the first compile-time array.

The start with a start-line and are terminated by an end-line.

These two lines will be generated by COS when the very first documentation entries are inserted.

The start-line starts with the character string **,/\$\$A** from position **8** (in all source types).

The end-line starts with the character string **,/\$\$Z** from position **8** (in all source types).

Lines marked with **,/\$\$B** hold heading text.

Lines marked with **,/\$\$Y** are blank lines.

Example for start- and end-lines:

0099.00	*/\$\$A	Do not change/remove data from this line down to end line \$\$A	010627
0100.00	*/\$\$B	COS S o u r c e H i s t o r y E n t r i e s	010627
0101.00	*/\$\$Y		010627
..			
..			
0200.00	*/\$\$Z	Do not change/remove data from this line up to start line \$\$Z	010627
*****Datenende *****			

Between start and end line there may be many various documentation entries (History Entries). The entries are in chronologic ascending order.

Each documentation entry

- Starts with a **Event-Start-Line**, which is identified by character string **,/\$\$C** from position **8** (in all source types) and
- Ends with a **Event-End-Line**, which is identified by character string **,/\$\$Y** from position **8** (in all source types).

The Event-Start-Line contains:

From position 13 to position 18 the Event Code

From position 20 to position 27 the Event Date

From position 29 to position 34 the Event Time

From position 36 to position 45 the Event User

From position 47 to position 66 Job name and Job number (JJJJJJJJJ/NNNNNN).

Chapter E.

The Object Creation Log File

If ***YES** was specified as value for COS keyword **WRTCRTLOGE**, a log entry is written into file **CRTOBJLOG** after an object was created successfully.

The file CRTOBJLOG must be found in the current job's library list. Therefore this file should be created in a common library (e.g. QGPL or QUSRSYS).

The log file CRTOBJLOG has following record structure:

Feldname	Feld- Bezeichnung	Daten- typ	Länge	Dez	Bemerkung
SRCMBR	SRC- MEMBER	A	10		Source file member
SRCTYPE	SRC-TYPE	A	10		Source type
SRCFILE	SRC-FILE	A	10		Source file
SRCLIB	SRC- LIBRARY	A	10		Library of source file
OBJLIB	OBJ- LIBRARY	A	10		Object library
SYSTEM	SYSTEM	A	8		System name
USER	CRT-USER	A	10		User of creation
DATE	CRT-DATE	S	8	0	Date of creation
TIME	CRT-TIME	S	6	0	Time of creation

The utilization of the data may be done individually. COS supplies in this regard no additional functions.

Kapitel F.

Software-Installation

Prerequisites for installation:

COS requires a RISC-based IBM System i/iSeries running under OS version **V5R4M0** or higher.

Installation instructions:

1. Download the Zip file COS.ZIP. When unzipped, there should be 3 files on your PC:

- This document in German and in English (PDF files)
- COS.SAV. This save file contains the source file COSSRC with all required source members for creating this tool.

2. Create the save file COS on your system in a library of your choice

e.g. **CRTSAVF FILE(QGPL/COS)**

3. Upload the save file from your PC to the System i/iSeries by using FTP:

Open an MS-DOS window on your PC.

Type **ftp** to start a File Transfer Protocol session.

Type **lcd c:\temp** (or whatever local directory contains the COS.SAV).

Type **open xxx.xxx.xxx.xxx** (where the **x**'s are the TCP/IP address of the iSeries).

Enter your user id and password.

Type **bin** to set the transfer mode to binary image.

Type **cd qgpl** (or any other library where you have created the save file COS in step 2 above) to change to the iSeries library.

Type **put cos.sav** to upload the save file.

Type **quit** to exit the FTP session.

(You may also use any other FTP method for transferring the save file in COS.SAV to your iSeries.)

4. Restore the source file COSSRC on your iSeries:

RSTOBJ OBJ(COSSRC) SAVLIB(COS) DEV(*SAVF)

SAVF(save file lib/COS) RSTLIB(source file lib)

save file lib = The library, where the save file COS is located.

source file lib= The library, where the source file should be restored to.

5. Restore the example data area TEST with a library list on your iSeries:

RSTOBJ OBJ(TEST) SAVLIB(COS) DEV(*SAVF)

SAVF(save file lib/COS) RSTLIB(obj lib)

save file lib = The library, where the save file COS is located.

obj lib = Library, where the objects are to be created.

You may rename and/or duplicate this data area in order to create your own individual "library list data areas".

6. Create the install program:

```
CRTBNDCL PGM(obj lib/CRT_COS)
          SRCFILE(source file lib/COSSRC)
```

7. Execute the install program:

```
CALL PGM(obj lib/CRT_COS) PARM('object lib' 'source file lib')
```

Parameters:

'obj lib' = Library, where the objects are to be created.
'source file lib' = Library, where the source file COS is
located (restored in step 4 above).

These libraries may be identical.

The program has ended when the message "COS Tool has been installed successfully" appears.

8. Good luck and have fun.

Author: Werner Noll
GEFIS Gesellschaft fuer Individual-Software mbH
Ludwigstr. 9, 63739 Aschaffenburg, GERMANY
Email: werner.noll@gefis.com.

Remarks: This is free software. You can redistribute it and/or modify it. The only restriction is that you may not distribute it for the purposes of monetary gain. We do not guarantee for the expected functionalities. The use of this software is always at your own risk.
All comments or requests for enhancements are welcome.