

COS

Create Objects Support

Programmer Guide

PG-COS

17. April 2009

Kapitel A.

COS – Konzept und Funktionen

Das Tool **COS** wurde von GEFIS zur Unterstützung der eigenen Softwareentwicklung geschaffen.

Es ist ein Tool für Programmierer und dient u.a.:

- Zur **Vereinfachung der Kompilierungs- und Bindevorgänge** durch **Standardisierung der Parameter in Erstellungsbefehlen**.
- Zur **Vermeidung von Fehlern**, die bei der Objekterstellung durch falsche oder unvollständige Parameterdefinitionen und/oder Bibliothekssuchlisten entstehen.
- Zur **Unterstützung der Änderungsdokumentation** durch maschinelle Dokumentationen in Quelldaten und Prokolldatei.

Mit COS wird die Erstellung von folgenden Objekten unterstützt:

- **ILE-RPG-Programme**
- **ILE-RPG-Serviceprogramme**
- **ILE-CL-Programme**
- **Display-Files**
- **Printer-Files**

Die Kernfunktionen von COS sind der Einsatz des COS-Befehls **CRTOBJCOS**, der an Stelle der entsprechenden OS-Befehle zur Erstellung von Objekten aus Quelldaten verwendet wird.

Die für die Funktionen von COS erforderlichen Schlüsselinformationen werden in den Quelldaten in Form von Pseudo-Code (genannt COS-Anweisungen) hinterlegt und vor Ausführung eines Erstellungsbefehls interpretiert.

Die Objekterstellung wird optional am Ende der Quelldaten als *Source History Entry* dokumentiert.

COS wurde unter OS Version **5**, Release **4** entwickelt und erfordert bei dessen Einsatz mindestens diesen Aktualitätsstand des Betriebssystems.

Kapitel B.

Erstellung von Objekten mit COS

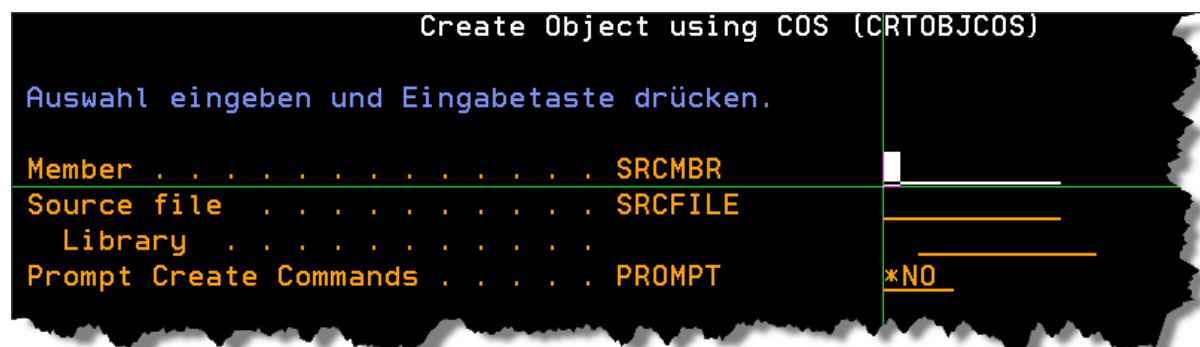
1. Der Befehl CRTOBJCOS (Create Object using COS)

Für die Erstellung von **ILE-Programmobjekten** und von **Dateiobjekten** vom Typ **DSPF** und **PRTF** ist zwingend und ausschliesslich der Befehl **CRTOBJCOS** zu verwenden. Die Ausführung des Befehls kann interaktiv oder im Batchmodus erfolgen.

Mit dem Befehlsparameter **PROMPT** kann bei interaktiver Ausführung des Befehls ein prompten der OS-Erstellungsbefehle (z.B. CRTRPGMOD, CRTPGM, etc.) erzwungen werden. Dies ist nur in Ausnahmefällen erforderlich, wenn spezielle Erstellungsoptionen erforderlich sind, die von COS nicht unterstützt werden.

Bei Ausführung des Befehls CRTOBJCOS im Batchmodus wird die Angabe im Befehlsparameter PROMPT ignoriert, da hier ein prompten nicht möglich ist.

Der Aufruf des Befehls CRTOBJCOS erfolgt i.d.R. über eine benutzerdefinierte Auswahl in der **PDM**-Arbeitsanzeige "Mit Teildateien arbeiten (mittels **PDM**)", z.B. mit der **PDM**-Auswahl '**CC**'.



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 with four rows of input fields. The first row is "Member SRCMBR" with a cursor in the SRCMBR field. The second row is "Source file SRCFILE" with a cursor in the SRCFILE field. The third row is "Library" with a cursor in the field. The fourth row is "Prompt Create Commands PROMPT" with a cursor in the PROMPT field. To the right of the PROMPT field, the text "*NO" is visible.

Nach Aufruf des Befehls erfolgen zunächst umfangreiche Plausibilitätsprüfungen. Im Fehlerfall werden entsprechende Nachrichten an den Benutzer gesendet.

Wenn im COS-Schlüsselwortparameter **WRTSRCHSTE** der Wert ***YES** spezifiziert wurde, werden nach der erfolgreichen Erstellung eines Objektes an das Ende der Quellenanweisungen bestimmte Erstellungsinformationen (Source History Entries) ausgegeben. Sie enthalten u.a. den für die Objekterstellung erzeugten und benutzten OS-Erstellungsbefehl mit allen Parametern.

Die Syntax dieser Dokumentationsdaten ist im u.g. **Kapitel D** detailliert beschrieben.

Wenn im COS-Schlüsselwortparameter **WRTCRTLOGE** der Wert ***YES** spezifiziert wurde, wird nach der erfolgreichen Erstellung eines Objektes ein Protokoll-Eintrag in die Datei **CRTOBJLOG** geschrieben. Die Datei CRTOBJLOG muß über die aktuelle Bibliothekssuchliste gefunden werden können. Eine Beschreibung der Logdatei enthält das u.g. **Kapitel E**.

2. Die COS Steueranweisungen in den Quellencodes

Um den COS-Befehl **CRTOBJCOS** benutzen zu können, müssen in den Quelldaten bestimmte steuernde Eintragungen vorgenommen werden. Diese Eintragungen sind im folgenden als **COS-Anweisungen** bezeichnet.

Bei der Ausführung der o.g. Befehle werden diese COS-Anweisungen hinsichtlich deren Existenz, Syntax und Plausibilität hin überprüft. Die eigentliche Befehlsausführung (=Objekterstellung) wird erst dann durchgeführt, wenn die COS-Anweisungen vollständig und fehlerfrei sind.

2.1. Allgemeine Regeln zu den COS-Anweisungen in Quellendateien

2.1.1. Start- und Ende-Anweisungen

Die COS-Anweisungen werden von den folgenden Beginn- und Ende-Kennzeichen eingeschlossen, die in Kommentarzeilen des Quellencodes **vor der ersten echten Codezeile** zu spezifizieren sind. In jeder Quellenteildatei darf eine COS-Gruppe nur einmal vorkommen.

```
/$B   = Start der COS-Anweisungen  
...   COS-Anweisung 1;  
...   COS-Anweisung 2;  
...   COS-Anweisung n;  
/$E   = Ende der COS-Anweisungen
```

2.1.2. Allgemeine COS Syntax

Jede COS-Anweisung (incl. der Start- und Ende-Anweisungen) kann **nur in einer Kommentarzeile** des Quellencodes dargestellt werden.

Daraus ergibt sich, daß die COS-Anweisungen in RPG- und DDS-Quellen **ab der Stelle 8** und in CLP-Quellen **ab der Stelle 3** einer Zeile beginnen können.

Die COS-Anweisung beginnt grundsätzlich mit einem COS-Schlüsselwort, dem eine Schlüsselwerte-Liste folgt, die **in runden Klammern** eingebettet ist.

Schlüsselwerte werden immer als alphabetische Zeichenkette betrachtet; die Einbettung in Hochkommata entfällt.

Eine Schlüsselwerte-Liste **endet** mit der runden Klammer, die das Ende der Werteliste markiert.

Eine COS-Anweisung (ausser /\$B und /\$E) endet immer mit einem ; (Semikolon).

Die **Reihenfolge** der COS-Anweisungen (zwischen der Start- und Ende-Anweisung) ist grundsätzlich beliebig.

Eine Werteliste kann in einer **Folgezeile ab Stelle 8 fortgesetzt** werden, ohne daß ein Fortsetzungszeichen spezifiziert wird. Das Ende der COS wird vom ; bestimmt.

Syntax-Beispiele im RPG-Quellencode:

```
* /$B
* Schlüsselwort1(Wert1);
* Schlüsselwort2(Wert1 Wert2);
* Schlüsselwort3(Wert1 Wert2 Wert3
* Wert4);
* /$E
```

Der Stern * auf Stelle 7 markiert die Zeile als Kommentarzeile. Die Stelle 8 wurde aus optischen Gründen leer gelassen.

Syntax-Beispiele im CLP-Quellencode:

```
/* /$B                                     */
/* Schlüsselwort1(Wert1 Wert2 Wert3);    */
/* Schlüsselwort2(Wert1);                */
/* Schlüsselwort3(Wert1 Wert2 Wert3 Wert4 */
/* Wert5);                               */
/* /$E                                     */
```

Die Zeichenfolge ,/' ab Stelle 1 oder höher markiert die Zeile als Kommentarzeile. Kommentarzeilen werden durch die Zeichenfolge ,*/' beendet. Kommentar-Fortsetzungszeilen können mit einem Plus ,+' am Zeilenende spezifiziert werden. Beispiel:

```
/* /$B                                     */
/* Schlüsselwort3(Wert1 Wert2 Wert3 Wert4 +
Wert5);                                   */
/* /$E                                     */
```

2.2. COS-Anweisungen im Quellencode

COS-Keyword	Key Value(s)	M= Zwingend	Valid Types
TYPE	COS-Objekt-Typ: RPG = ILE-RPG-Modul aus Source Types RPGLE und SQLRPGLE. RPGSRV = ILE-RPG-Serviceprogram-Modul aus Source Types RPGLE und SQLRPGLE. CLP = ILE-CLP Modul aus Source Type CLLE. DSPF = Displayfile PRTF = Printerfile	M	ALL
OBJLIB	Name der Objektbibliothek für die Objekttypen *PGM und *FILE	M	ALL
WRTSRCHSTE	Source-History-Entry ausgeben: *YES oder *NO (siehe Kapitel D)		ALL
WRTCRTLOGE	Eintrag in Erstellungs-Protokolldatei CRTOBJLOG : *YES oder *NO . Die Datei CRTOBJLOG muß über die aktuelle Bibliothekssuchliste gefunden werden können. (siehe Kapitel E)		ALL
LIBLARA	Datenbereich der Bibliothekssuchliste: Datenbereichsname oder Bibliothekssname/Datenbereichsname Hinweis: Die Bibliothekssuchlisten in den Datenbereichen können mit dem Befehl EDTLIBLDTA editiert werden (siehe Kapitel C).		ALL
ACTGRP	Aktivierungsgruppe entsprechend Parameterwerte aus den Befehlen CRTPGM und CRTSRVPGM: Programmname oder QILE oder *CALLER	M	RPG RPGSRV CLP
OPTION	Entsprechende Parameterwerte aus den Befehlen CRTPGM und CRTSRVPGM, z.B. OPTION(*DUPPROC *DUPVAR)		RPG RPGSRV CLP
USRPRF	Entsprechende Parameterwerte aus den Befehlen CRTPGM und CRTSRVPGM, z.B. USRPRF(*OWNER)		RPG RPGSRV CLP
BNDSRVPGM	Entsprechende Parameterwerte aus dem Befehl CRTPGM. (Bei RPG-Programmen werden Serviceprogramme grundsätzlich über Binderverzeichnisse benutzt)		CLP
BNDDIR	Entsprechende Parameterwerte aus dem Befehl CRTPGM. (Bei RPG-Programmen werden Binderverzeichnisse auf den H-Specifications geholt)		CLP
EXPORT	Entsprechender Parameterwert aus dem Befehl CRTSRVPGM, Z.B.: EXPORT(*SRCFILE) Wird hier *SRCFILE spezifiziert, wird als Name der Exportquellendatei der Unterlassungswert QSRVSRC angenommen.	M	RPGSRV

COS-Keyword	Key Value(s)	M= Zwingend	Valid Types
OVRDBF	CL-Befehl zum Überschreiben einer Datenbankdatei zum Zeitpunkt der Modulerstellung (ohne Parameter OVRSCOPE). Beispiel: OVRDBF FILE(FILEX) TOFILE(LIBY/FILEY) Die Löschung der Überschreibung erfolgt automatisch nach der Objekterstellung.		RPG RPGSRV CLP
LVLCHK	Satzformat-Aktualitätsprüfung (*NO, *YES). Beispiel: LVLCHK(*NO) (Unterlassungswert ist *NO)		DSPF PRTF
DEVTYPE	Entsprechender Parameterwert aus dem Befehl CRTPRTF, Z.B.: DEVTYPE(*IPDS)		PRTF
PAGESIZE	Entsprechender Parameterwert aus dem Befehl CRTPRTF, Z.B.: PAGESIZE(72 132) (Unterlassungswert ist 72 132) Als dritter Schlüsselwortparameter wird immer der Unterlassungswert * ROWCOL angenommen.		PRTF
OVRFLW	Entsprechender Parameterwert aus dem Befehl CRTPRTF, Z.B.: OVRFLW(66) (Unterlassungswert ist 66)		PRTF
HOLD	Entsprechender Parameterwert aus dem Befehl CRTPRTF, Z.B.: HOLD(*YES) (Unterlassungswert ist *YES)		PRTF

2.3. Codierbeispiele für COS-Anweisungen

RPG-Programm

```
0001.00 *****
0002.00 * GEFIS GESELLSCHAFT FÜR INDIVIDUAL-SOFTWARE MBH *
0003.00 *****
0004.00 * Program : EXAMPLE1 *
0005.00 * Function: RPG-Programm-Beispiel *
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-Programm

```
0001.00 *****
0002.00 * GEFIS GESELLSCHAFT FÜR INDIVIDUAL-SOFTWARE MBH *
0003.00 *****
0004.00 * Program : EXAMPLE2 *
0005.00 * Function: RPG-Service-Programm-Beispiel *
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-Programm

```
0001.00 /*****
0002.00 /* GEFIS GESELLSCHAFT FÜR INDIVIDUAL-SOFTWARE MBH */
0003.00 /*****
0004.00 /* Program : EXAMPLE3 */
0005.00 /* Function: CL-Programm-Beispiel.. */
0006.00 /* Type : CLLE */
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: Displayfile-Beispiel *
* Author   : *
*****
*/$B
* TYPE(DSPF);
* OBJLIB(GEFISWN);
* LVLCHK(*NO);
*/$E
*
```

Printer-File

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

Kapitel C.

Verwalten der Bibliothekssuchlisten

Bei der Erstellung von Programmobjekten wird in COS eine Bibliothekssuchliste verwendet, die über die COS-Anweisung **LIBLARA** festgelegt wird. Im Schlüsselwort-Wert dieser COS-Anweisung muß (qualifiziert oder unqualifiziert) ein Datenbereich angegeben werden, der eine gültige Bibliothekssuchliste enthält.

Die maximale Länge eines solchen Datenbereichs beträgt 2000 Bytes (= 181 Bibliotheken in Liste).

Die Inhalte der Datenbereiche (= Bibliothekssuchlisten) können mit dem Befehl **EDTLIBLDTA** verwaltet werden. Dieser Befehl entspricht von der Benutzeroberfläche her weitgehend dem OS-Befehl EDTLIBL.

Beispiel:

```
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	

Weitere ...

F5=Refresh F3=Exit

Hinweis: Der COS-Befehl **CRTOBJCOS** prüft vor Ausführung der Objekterstellung, ob die Einträge in der Bibliothekssuchliste (noch) gültig sind.

Kapitel D.

Syntax der Source History Entries im Quellencode

COS unterstützt die Dokumentation durch maschinelle Fortschreibung von Dokumentationsinformationen als Kommentare in den jeweiligen Quelldaten.

Kommentarzeilen sind in

- **RPGLE-**, **DSPF-** und **PRTF-**Quelldaten durch einen Stern (*) auf Stelle **7** und in
- **CLLE-**Quelldaten durch die Zeichenfolge **,/*'** ab der Stelle **1** und die Zeichenfolge **,/*'** am Ende des Kommentars gekennzeichnet. Das Ende des Kommentars in CLLE-Quelldaten könnte auch in einer Fortsetzungszeile stehen, die durch ein Pluszeichen in der vorherigen Zeile als solche erkenntlich ist. Die von COS generierten Kommentarzeilen in CLLE-Quelldaten sind jedoch in jeder Zeile ab Stelle 79 mit der Zeichenfolge **,/*'** begrenzt.

Alle Dokumentationseinträge befinden sich am Ende der Quelldaten (nach der letzten Quellenzeile). Sie beginnen mit einer Start-Zeile und werden mit einer Ende-Zeile begrenzt. Diese Zeilen werden beim ersten Dokumentationseintrag maschinell generiert.

Die Start-Zeile beginnt mit der Zeichenfolge **,/\$\$A'** ab Stelle **8** (bei allen Quellenarten).
Die Ende-Zeile beginnt mit der Zeichenfolge **,/\$\$Z** ab Stelle **8** (bei allen Quellenarten).
Die Zeile mit der Kennzeichnung **,/\$\$B'** ist ein Hinweistext.
Die Zeile mit der Kennzeichnung **,/\$\$Y'** ist eine Leerzeile.

Beispiel für Start- und Ende-Zeilen:

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*****			

Zwischen der Start- und Ende-Zeile können beliebig viele Dokumentationseinträge (History Entries) vorkommen. Die Einträge sind in chronologisch aufsteigender Folge.

Jeder Dokumentationseintrag

- beginnt mit einer **Event-Start-Zeile**, welche durch die Zeichenfolge **,/\$\$C'** an Stelle **8** (bei allen Quellenarten) gekennzeichnet ist und
- endet mit einer **Event-Ende-Zeile**, welche durch die Zeichenfolge **,/\$\$Y'** ab Stelle **8** (bei allen Quellenarten) gekennzeichnet ist.

Die Event-Start-Zeile enthält:

Von Stelle 13 bis Stelle 18 den Event-Code

Von Stelle 20 bis Stelle 27 das Event-Datum

Von Stelle 29 bis Stelle 34 die Event-Uhrzeit

Von Stelle 36 bis Stelle 45 den Event-Benutzer

Von Stelle 47 bis Stelle 66 den Jobnamen und die Jobnummer (JJJJJJJJJ/NNNNNN).

Mögliche Event-Codes sind:

CRTMOD Eintrag bei Modulerstellung
CRTPGM Eintrag bei Programmerstellung
CRTDSP Eintrag bei Displayfile-Erstellung
CRTPR Eintrag bei Printerfile-Erstellung

Danach erscheint die **Subject-Zeile**, welche durch die Zeichenfolge **/\$\$D** ab Stelle 8 gekennzeichnet ist.

Sie enthält

von Stelle 13-22: Den Namen des Quellentyps (z.B. RPGLE)

von Stelle 52-59 den Systemnamen der Objekterstellung

Danach folgen in beliebig vielen Textzeilen die vollständigen Objekterstellungsbefehle, die u.a. die Bibliotheksnamen der Quelldaten und des Objektes enthalten.

Beispiel für die verschiedenen Dokumentationseinträge in einer RPGLE-Quelle:

0099.00	*/\$\$A Do not change/remove data from this line down to end line \$\$Z	010627
0100.00	*/\$\$B COS source History Entries	010627
0101.00	*/\$\$Y	010627
0017.00	*/\$\$C CRTMOD 20080502 093810 GEFISWN QPADEV0006/085348	010627
0118.00	*/\$\$D RPGLE PETRIAS1	010627
0119.00	* CRTRPGMOD MODULE(MS00S1/MSR500) SRCFILE(MS00S1/QRPGLESRC) +	010627
0120.00	* DBGVIEW(*LIST)	010627
0121.00	*/Y	010627
0022.00	*/\$\$C CRTPGM 20080502 093906 GEFISWN QPADEV0006/085348	010627
0123.00	*/\$\$D RPGLE PETRIAS1	010627
0124.00	* CRTPGM PGM(MS00P1/MSR500) MODULE(MS00S1/MSR500) ACTGRP(QILE) +	010627
0125.00	* OPTION(*DUPPROC *DUPVAR)	010627
0126.00	*/\$\$Y	010627
0127.00	*/\$\$Z Do not change/remove data from this line up to start line \$\$A	010627
	*****Datenende *****	

Kapitel E.

Die Objekterstellungs-Protokolldatei

Wenn im COS-Schlüsselwortparameter **WRTCRTLOGE** der Wert ***YES** spezifiziert wurde, wird nach der erfolgreichen Erstellung eines Objektes ein Protokoll-Eintrag in die Datei **CRTOBJLOG** geschrieben. Die Datei CRTOBJLOG muß über die aktuelle Bibliothekssuchliste gefunden werden können. Deshalb sollte sie vorzugsweise in einer allgemeinen Bibliothek (z.B. QGPL oder QUSRSYS) verwaltet werden.

Die Protokolldatei CRTOBJLOG hat folgenden Satzaufbau:

Feldname	Feld-Bezeichnung	Daten-typ	Länge	Dez	Bemerkung
SRCMBR	SRC-MEMBER	A	10		Quellenteildatei
SRCTYPE	SRC-TYPE	A	10		Quellentyp
SRCFILE	SRC-FILE	A	10		Quellendatei
SRCLIB	SRC-LIBRARY	A	10		Bibliothek der Quellendatei
OBJLIB	OBJ-LIBRARY	A	10		Objekt-Bibliothek
SYSTEM	SYSTEM	A	8		Systembame
USER	CRT-USER	A	10		Benutzer der Erstellung
DATE	CRT-DATE	S	8	0	Datum der Erstellung
TIME	CRT-TIME	S	6	0	Uhrzeit der Erstellung

Die Nutzung der Protokolleinträge in dieser Datei kann individuell erfolgen. COS hat diesbezüglich keine Funktionen.

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.