

## CPG-Shell

Mit CPG-Shell kann man aus einem CICS- oder Batchprogramm aus z/VSE oder z/OS eine Anwendung auf einer anderen Plattform starten und beliebig viele Parameter übergeben. In der aktuellen Version ist der Übergabebereich auf 2.000 Bytes begrenzt.

Mit CPG-Shell kann man Prozesse auf anderen Systemen zu einer bestimmbaren Zeit ablaufen lassen – aus jeder Anwendung im CICS und im Batch.

### Beispiel für Onlineverarbeitung:

Eine Anwendung schreibt Informationen nicht nur auf dem Host in eine VSAM-Datei, sondern stößt gleichzeitig eine Verbuchung genau dieser Daten mit einer beliebigen Anwendung auf einer anderen Plattform an.

CPG-Shell kann von jeder Programmiersprache online mit einem EXEC CICS LINK aufgerufen werden und auf der Serverplattform einen Shell-Befehl ausführen. Der Shell-Befehl wird als Parameter über die Common Area übergeben und kann der Aufruf eines Programms (.exe), ein Befehlsskript (.cmd oder .bat) oder der Aufruf einer Java-Klasse sein. Auch diesem Befehl können Parameter mitgegeben werden.

### Beispiel für Batchverarbeitung:

Eine Anwendung überträgt via FTP eine Host-Datei auf einen Server und ruft anschließend eine PC-Anwendung auf, die die übergebene Datei auf der PC-Seite in eine Datenbank stellt.

Für Batchverarbeitung stellt CPG-Shell eine Call-Schnittstelle zur Verfügung, sodass Parameter mit jeder Programmiersprache übergeben werden können. CPGSHBAT ist ein Batchprogramm, das über SYSIPT Befehle an die Serverkomponente übergibt und vom z/VSE aus Programme auf einer anderen Plattform ausführen kann.

Das Produkt CPG-Shell besteht aus mehreren Komponenten. Die wichtigsten sind:

- der Client-Aufruf, der auf einem z/VSE oder z/OS installiert wird und
- die Serverkomponente, die z.B. auf Windows oder Linux installiert wird.

Damit man Parameter wie z.B. den Pfad zu Programmen oder Befehlsdateien flexibel gestalten kann, können Befehle in einem Programm QSHELLTB abgelegt und über ein achtstelliges Schlüsselwort in die Parameterübergabe eingebunden werden. In der Common Area wird das Schlüsselwort angegeben, um serverseitig das Programm zu starten.

Parameter können auch ohne QSHELLTB über die Common Area übergeben werden, jeweils durch ein Blank getrennt. Werden Pfadangaben gemacht, die Blanks enthalten, so muss der Pfad in Hochkommata eingeschlossen werden.

## Aufruf des Programms CPGSHELL

### 1. Variante: Parameterübergabe vollständig über Common Area

Das Programm CPGSHELL wird im CICS-Programm über einen EXEC CICS LINK aufgerufen.

Hierbei sehen die Parameter z.B. wie folgt aus (in CPG):

```

- FIELD CPGCOM TYPE WORDST.
-      BLANK 250.
-      BLANK 500.
-      BLANK 750.
-      BLANK 1000.
-      BLANK 1250.
-          10 '$$$$$$$$00'
-          29 'c:\winword.bat '
-          118 '000.000.'
-          139 'nnn.nnn.nnn.nnn:ppppp'
- * .....+.....1.....+.....2.....+.....3.....+.....
-      USRIPA 133
-          149 '0'. * NO RETURNCODE
-          150 'C'
-      PARM1 251
    
```

1-8	\$\$\$\$\$\$\$\$	Schlüsselwort für „Kein QSHELLTB-Eintrag“ = Alle Parameter stehen in der Common Area
9-10	00	Die SYSID der TCP/IP Partition
11-110		Aufruf Befehl des Serverprogramms. Hier beispielsweise: "c:\winword.bat "
111-118	000.000.	Start der IP-Adresse
119-139	nnn. nnn. nnn. nnn:ppppp	IP-Adresse und Port. Alle Stellen müssen drei- stellig eingetragen werden, der IP-Port fünfstellig.
140-148		Reserviert
149	0,1,2	0 = kein Return Code, asynchroner Aufruf 1 = warten bis fertig. CICS-Anwendung ist aktiv solange die PC-Anwendung aktiv ist. Der Return Code steht in den ersten beiden Stellen der Common Area, im Fehlerfall ist er 'EF'. 2 = alle Meldungen an den Host senden. Verarbei- tung wie 1, aber alle Meldungen, die die PC- Anwendung auf die DOS-Konsole schreibt, stehen nach der Verarbeitung im CICS in der TS-Queue mit dem Namen „gepackte Task- nummer“+QSHL zur Verfügung.
150	C	Blank: <b>Default ist der Eintrag '2'</b> Konsolmeldung ausgeben
151-2100		weitere Parameter, jeweils 150 Bytes

## 2. Variante: Parameterübergabe über QSHELLTB

Es gibt noch eine weitere Möglichkeit, CPG-Shell zu starten. Dabei wird ein Schlüsselwort in die ersten acht Stellen der Common Area gesetzt. Dies bedeutet, dass im Programm QSHELLTB dieses Schlüsselwort gesucht wird (im Beispiel unten: QITSTART) und entsprechend die zugehörigen Parameter aus QSHELLTB geladen werden.

Als Beispiel soll der folgende Aufruf (in COBOL) dienen:

```
MOVE 83 TO MYLEN.  
  
EXEC CICS LINK  
  PROGRAM( , 'CPGSHELL' )  
  COMMAREA(MYCOM)  
  LENGTH(MYLEN)  
END-EXEC
```

Die Common Area wird im COBOL-Programm wie folgt definiert (die Bedeutung der Parameter ist auf der folgenden Seite unter „zugehörige Common Area“ erklärt):

```
01 MYPARM.  
  02 MYCOM          PIC X(0008)  VALUE 'QITSTART'.  
  02 MYSTACK        PIC X(2)      VALUE ' '.  
  02 MYTYPE         PIC X(5)      VALUE '/C=P '.  
  02 MYFILE         PIC X(12)     VALUE '/F=QPGASTAT '.  
  02 MYCLASS        PIC X(6)      VALUE '/CL=Q '.  
  02 MYUSR          PIC X(8)      VALUE '/U=UID '.  
  02 MYPWD          PIC X(7)      VALUE '/P=PWD '.  
  02 MYLOG          PIC X(4)      VALUE 'LOG '.  
  02 MYFCH          PIC X(4)      VALUE 'FCH '.  
  02 MYDEL          PIC X(4)      VALUE 'DEL '.  
  02 MYOUTPATH      PIC X(22)     VALUE '/O=F:\TestPdf\Source\ '.  
  02 MYPARM-END     PIC X(1)      VALUE ' '.  
  
01 MYLEN           PIC S9(4).
```

Über den Command EXEC CICS LINK werden alle Parameter übergeben,

- Zunächst die Parameter, die in QSHELLTB stehen,
- dann die Parameter, die im Programm zusätzlich eingetragen sind.

## QSHELLTB

```

*-----*
*NAME      DC      CL8 'NAME'                ANKERNAME FÜR DEN AUFRUF      *
*SYSID     DC      CL2 '00'                SYSID FÜR TCP/IP IM Z/VSE     *
*PROGRAM   DC      CL50 'C:\\WINDOWS\\notepad.exe '          *
*P2        DC      CL150 ' '                2. Teil des Namens          *
*IP        DC      CL29 '000.000.nnn.nnn.nnn.nnn:ppppp' IP+Port      *
*RESERVE   DC      CL10 'C'C'              Reserved, Consolemessages    *
*****1.....2.....3.....4.....5*
QITSTART   DC      CL8 'QITSTART'          QIT-AUFRUF                    *
           DC      CL2 '00'                TCP/IP_SYSID                  *
           DC      CL50 'D:\\Programme\\Lattwein\\QIT\\qit.exe ' *
           DC      CL50 ' '                *
           DC      CL29 '000.000.nnn.nnn.nnn.nnn:ppppp'      *
           DC      CL10 ' ',C'C'          RESERVED,CONSOLMESSAGES     *
*****1.....2.....3.....4.....5*
    
```

In der Tabelle QSHELLTB wird der Eintrag für QITSTART in etwa so aussehen wie hier beschrieben.

- 1 - 8        Schlüsselwort
- 9- 10      00 Ist die SYSID der IP Partition (Stack) im z/VSE
- 11- 110    100 Stellen für den Aufruf des Serverprogramms. Hier wird in der Regel der vollständige Pfad und Programmname bzw. der Name einer .bat oder .cmd Datei eingetragen.
- 111-139    IP- Adresse und Port des Servers, auf dem CPG-Shell gestartet wurde.
- 140-148    reserviert
- 149        Verarbeitungsart 0,1,2. Siehe oben, Parametere Aufbau unter „1.Variante“
- 150        C für Konsolmeldung CPG-Shell gestartet

Jeder Tabelleneintrag ist 150 Stellen lang.

## Zugehörige Common Area

```

.....1.....2.....3.....4.....5.....6.....7.....+
QITSTART /C=P /F=QPGASTAT /CL=Q /U=UID /P=PWD LOG FCH DEL
/O=F:\TestPdf\Source\ '
    
```

- 1-8        QITSTART            Schlüsselwort für QSHELLTB-Eintrag (s.u.)
- 11-100    Die Parameter, die für einen Aufruf von QIT gebraucht werden.
  - /C=P                    POWER
  - /F=QPGASTAT            Name der Powerliste
  - /CL=Q                   Power List Klasse Q
  - /U=UID                   User
  - /P=PWD                   Passwort
  - LOG                    Logdatei mitschreiben
  - FCH                    File Check, vorhandene Dateien um Nummer erweitern
  - DEL                    Löschen nach Übertragung
  - /O=F:\TestPdf\Source\    Ausgabepfad

## CPG-Shell auf dem PC

Auf dem PC muss ein JRE (Java Runtime Environment) installiert sein. Dieses sollte mindestens die Version 1.6 haben. CPG-Shell wird in ein Verzeichnis entpackt und mit dem folgenden Konsolbefehl gestartet:

```
java -jar CPGShell.jar
```

Folgende Meldungen werden im DOS-Fenster angezeigt:

```
C:\Programme\Lattwein\CPGShell >java CPGShell
C:\Programme\Lattwein\CPGShell \CPGShell.properties
Got properties from CPGShell.properties
-- listing properties --
MESSAGES=ON
PARMFILE=C:\Parm01.dat
PORT=pppp
TRACELEVEL=0
CPGShell gestartet auf Port pppp
ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=pppp]

External Reader gestartet
```

Dies sollte in dem Verzeichnis aufgerufen werden, in dem CPG-Shell entpackt wurde. Daher wird im Classpath der aktuelle Pfad erwartet und sollte wie folgt angelegt worden sein:

```
set Classpath=.;%classpath%
```

Im Installationspfad gibt es die **Property-Datei für CPG-Shell**. Die Properties sehen wie folgt aus:

```
#####
#CPGShell Properties
#####
# Port on which the Db2Daemon listens
#####
PORT=pppp
#print messages (on) or do not print (off)
MESSAGES=ON
#enter the tracelevel, 0 no trace, 1 normal, 2 is extended
TRACELEVEL=0
#####
PARMFILE=C:\\Parm01.dat
#####
```

PORT=pppp gibt den Port an, auf dem CPG-Shell lauscht, um Befehle vom Host entgegen zu nehmen.

MESSAGES = ON bedeutet, dass Meldungen in das aktuelle Konsolfenster geschrieben werden können.

TRACELEVEL=0 sollte immer auf 0 gesetzt sein.

PARMFILE bedeutet, dass die Parameter, die vom Host gesendet werden, in eine PC-Textdatei geschrieben werden und diese Datei als Parameter an das aufgerufene Programm übergeben wird. Wird der Parameter nicht angegeben, so wird der Programmaufruf mit der Parameterliste durchgeführt.

Dabei kann die Parameterliste sich wie folgt ändern:

Aufruf vom HOST:

```
C:\Aufruf01.bat "DAS#IST      #EIN      #BEISPIEL"
```

Ausgeführt wird aber:

```
C:\Aufruf01.bat "DAS#IST #EIN #BEISPIEL"
```

Mehrere Blanks werden als **ein** Blank dargestellt.

Dieses Problem kann man umgehen, indem man für die Blanks ein anders Zeichen wählt, dass dann vom aufgerufenen Programm wieder als Blank ersetzt wird. Also z.B.

```
C:\Aufruf01.bat "DAS#IST_____#EIN_____#BEISPIEL"
```

Oder man gibt den PARMFILE in der Property Datei an.

Dann sollte der Aufruf lauten:

```
C:\Aufruf01.bat C:\Parm01.dat
```

## CPG-Shell als Dienst starten

CPG-Shell kann als Dienst gestartet werden. Dazu kann man das Programm JavaService von der VSE Download Page der IBM herunterladen. Mit dem Befehl, der einmal aufgerufen wird, wird der Dienst etabliert.

```
JavaService.exe create CPGShell C:\Programme\Lattwein\CPGShell  
C:\Programme\Lattwein\CPGShell CPGShell
```

```
JavaService.exe create <ServiceName> <rootdir> <classpath>  
<mainclass> [<userid> <password>]
```

Den Download für JavaService findet man hier:

<http://www-03.ibm.com/systems/z/os/zvse/downloads/tools.html#javaservice>

## CPGSHBAT

CPGSHBAT ist ein Batchprogramm, das CPG-Shell benutzt und aus dem Batch Befehle an den CPG-Shell Listener sendet, der dann die entsprechenden Programme oder Befehlsdateien aufruft und ausführt. Ein Aufruf kann mehrere Befehle nacheinander ausführen. Dabei werden die Parameter über SYSIPT variabel an die aufgerufenen Programme, Befehlsdateien oder Java-Klassen weitergegeben.

### Beispiel:

Ein Kunde hatte die Anforderung, eigenen Quellcode, der in verschiedenen ICCF-Bibliotheken abgestellt ist, in einer Browseranwendung schnell und einfach nach beliebigen Schlüsselwörtern zu durchsuchen.

Der folgende Job lädt dazu täglich nachts die Daten und Indizes ausgewählter ICCF Libraries und stellt sie in einer MySQL-Datenbank ab:

```
// JOB IMPORT ICCF TO MYSQL
// EXEC CPGSHBAT,PARM='IPADDR=010.238.002.045:04799,IPPORT=04799'
$$$$$$$00ICCF_Import.exe PROD S:\Temp\ICCF\ICCF_INDEX_Clean_P.txt      *
S:\Temp\ICCF\ICCF_DATA_Clean_P.txt
/*
```

Hier wird also ein selbst entwickeltes Programm ICCF\_Import.exe aufgerufen, dem als Parameter die Positionen von zwei TXT-Files übergeben werden. Der dritte Parameter (PROD oder TEST) gibt an, von welchem VSE die Input-Daten abgeholt werden

CPG-Shell .....	1
Aufruf des Programms CPGSHELL.....	2
1. Variante: Parameterübergabe vollständig über Common Area.....	2
2. Variante: Parameterübergabe über QSHELLTB .....	3
QSHELLTB.....	4
Zugehörige Common Area .....	4
CPG-Shell auf dem PC.....	5
CPG-Shell als Dienst starten.....	6
CPGSHBAT .....	7