Fujitsu Software BS2000 CAPRI

Version 21.0
Juni 2025

Readme-Datei

Alle Rechte vorbehalten, insbesondere gewerbliche Schutzrechte. Änderung von technischen Daten sowie Lieferbarkeit vorbehalten. Haftung oder Garantie für Vollständigkeit, Aktualität und Richtigkeit der angegebenen Daten und Abbildungen ausgeschlossen. Wiedergegebene Bezeichnungen können Marken und/oder Urheberrechte sein, deren Benutzung durch Dritte für eigene Zwecke die Rechte der Inhaber verletzen kann.

Copyright © 2025 Fujitsu

Die Marke Fujitsu und das Fujitsu Logo sind Marken oder registrierte Marken von Fujitsu Limited in Japan und in anderen Ländern.

# 1 Ziel

Das Subsystem CAPRI soll die alte SVC79-Schnittstelle in der heutigen Form ablösen. Die neue Schnittstelle zeichnet sich dadurch aus, dass die Trennung von TU-Teil und TPR-Teil gegeben ist (keine Binder-Probleme). Die neue Schnittstelle zum TPR-Teil wird so angeboten, dass die DSL-Schnittstelle komplett versorgt ist und dadurch eine wesentlich vereinfachte Programmierung möglich ist.

# 2 Kurzbeschreibung Funktionalität

Durch die Unterstützung neuer Prozessoren gibt es immer mehr Probleme bei der Erstellung von SVC79-Programmen. Diese Programme müssen in mehrfacher Ausprägung und mit dem richtigen Assembler übersetzt angeboten werden.

CAPRI bietet 5 unterschiedliche Schnittstellen an:

1.) Ein Kommandointerface zum Starten von TPR-Routinen, die nur Eingabedaten benötigen.
2.) Eine Kommandoschnittstelle zum Laden von TPR-Modulen, die über den SVC79 angesprungen werden.
3.) Eine neue SVC79-Schnittstelle, die eigenständig TPR-Module nachladen kann.
4.) Den direkten Aufruf einer Systemroutine ohne TPR-Modul
5.) Absetzen eines TPR-SVC aus einer TU-Routine

Der SVC79 ist nicht mehr an die Kennung $TSOS gebunden, sondern kann zusätzlich über ein GUARD verwaltet werden. Der Systemverwalter kann die SVC79-Funktionen für andere Userids frei vergeben. Ist kein GUARD vorhanden, so ist das Verhalten entsprechend der CLASS2-Option SVC79. Die Kennung $SERVICE wird an der neuen Schnittstelle nicht unterstützt. Die alte SVC79 Schnittstelle ist weiterhin funktionsfähig.

# 3 Systemvoraussetzungen

Die erweiterten Zugriffe von Userids ungleich TSOS über die GUARD-Schnittstelle sind erst ab SECOS V4.0B möglich.

# 4 Kommandoschnittstelle

## 4.1 Direkte Programmausführung START-TPR-PROGRAM

```
------------------------------------------------------------------------
| START-TPR-PROGRAM                                                    |
------------------------------------------------------------------------
| FROM-FILE=*LIBRARY-ELEMENT(..)                                       |
|           |                                                          |
|           |  LIBRARY=*STD | <filename>                               |
|           |                                                          |
|           |  ,SYMBOL= <composed-name 1..8 with underscore>           |
|           |                                                          |
|          *BY-IMON(..)                                                |
|           |                                                          |
|           |  LOGICAL-ID=SYSLNK | <composed-name 1..30>               |
|           |                                                          |
|           |  ,INSTALLATION-UNIT=<composed-name 1..30>                |
|           |                                                          |
|           |  ,VERSION=<version 3..7> | *STD                          |
|           |                                                          |
|           |  ,DEFAULT-LIBRARY=<filename>                             |
|           |                                                          |
|           |  ,SYMBOL=<composed-name 1..8 with underscore>            |
|                                                                      |
|                                                                      |
|,PROGRAM-PARAMETERS=*NONE                                             |
|                    | <c-string  1..64>                               |
|                    | <x-string  1..128>                              |
|                    | <alpha-num 1..64>                               |
|                                                                      |
|,DIAGNOSIS-STOP=*NO | *AFTER-LOAD                                     |
|                                                                      |
------------------------------------------------------------------------
```

Mit diesem Kommando wird eine Routine in den privilegierten CL4-Systemspeicher geladen und ausgeführt. Die Routine kehrt zum Kommando zurück und wird entladen. CAPRI gibt die Returncodes auf SYSOUT aus. Ist dieser Entry über das LOAD-TPR-PROGRAM Kommando vorgeladen, so wird dieser Entry angesprungen und das Modul wird nicht entladen.

Beschreibung der Operanden

LIBRARY
 Name der Ladebibliothek, in der die Routine mit dem spezifizierten Entry steht. Ist der Name *STD, so wird die CAPRI-Ladebibliothek verwendet.

SYMBOL
 Entry oder Csect einer TPR-Routine.

LOGICAL-ID
 Der logische Name, der einer bestimmten Datei über IMON zugeordnet ist. Bei Modulbibliotheken ist die ID im allgemeinen SYSLNK.

```
INSTALLATION-UNIT
 Der Name des von IMON installierten Produktes.

VERSION
 Die Version des von IMON installierten Produktes. *STD ist die aktuelle Version

DEFAULT-LIBRARY
 Ist das Produkt nicht von IMON installiert, so wird dieser Name als Bibliotheks-
 Name genommen.

PROGRAM-PARAMETERS
 Dieses Feld ist 64 Zeichen lang und wird der TPR-Routine ungeprüft übergeben.

DIAGNOSIS-STOP= *NO / *AFTER-LOAD
 Mit diesem Parameter ist es möglich, dass die Routine geladen wird und der
 Start mittels einer beantwortbaren Meldung verzögert wird. Dadurch wird
 eine Diagnose mit den Testhilfen HELGA, IDIAS oder DAMP ermöglicht.
 Die Routine startet erst nach der Beantwortung der Frage CAP0004.
 Beim Testen ist zu beachten, dass diese Routinen bei jedem Aufruf in einen
 eigenen Kontext geladen werden und somit diese Entries mehrfach im System
 vorhanden sein können. Der Diagnoseschalter verhindert zusätzlich das
 Freigeben der Parameterareas bei abnormaler Termination.
```

## 4.2 Laden und Entladen von TPR-Modulen

Diese Kommandos sind nur zur Optimierung der Zugriffe nötig. Die Programme können vorgeladen werden. Alle Aufrufe mit dem angegebenen Symbol nutzen nur dieses TPR-Programm. Somit wird der Speicherplatz auf dieses Programm beschränkt. Außerdem fällt das dynamische Nachladen weg.

### 4.2.1 LOAD-TPR-PROGRAM

```
------------------------------------------------------------------
| LOAD-TPR-PROGRAM                                               |
------------------------------------------------------------------
| FROM-FILE=*LIBRARY-ELEMENT(..)                                 |
|           |                                                    |
|           |  LIBRARY=*STD | <filename>                         |
|           |                                                    |
|           |  ,SYMBOL= <composed-name 1..8 with underscore>     |
|           |                                                    |
|           *BY-IMON(..)                                         |
|           |                                                    |
|           |  LOGICAL-ID=SYSLNK | <composed-name 1..30>         |
|           |                                                    |
|           |  ,INSTALLATION-UNIT=<composed-name 1..30>          |
|           |                                                    |
|           |  ,VERSION=<version 3..7> | *STD                    |
|           |                                                    |
|           |  ,DEFAULT-LIBRARY=<filename>                       |
|           |                                                    |
|           |  ,SYMBOL=<composed-name 1..8 with underscore>      |
|                                                                |
------------------------------------------------------------------
```

Das Kommando lädt aus der angegebenen Bibliothek ein Modul mit dem
angegebenen Symbol in den CL4-Adressraum. Jeder Ladevorgang bekommt
einen eigenen Kontext. Es wird nur mit dem Privileg TSOS akzeptiert.

Beschreibung der Operanden

LIBRARY
Name der Ladebibliothek in der die Routine mit dem spezifizierten Entry
steht. Ist der Name *STD, so wird die CAPRI-Ladebibliothek verwendet.

SYMBOL
Entry oder Csect einer TPR-Routine.

LOGICAL-ID
Der logische Name, der einer bestimmten Datei über IMON zugeordnet ist.
Bei Modulbibliotheken ist die ID im allgemeinen SYSLNK.

INSTALLATION-UNIT
Der Name des von IMON installierten Produktes.

VERSION
Die Version des von IMON installierten Produktes. *STD ist die aktuelle Version.

DEFAULT-LIBRARY
Ist das Produkt nicht von IMON installiert, so wird dieser Name als Bibliotheks-
Name genommen.

### 4.2.2 UNLOAD-TPR-PROGRAM

```
| UNLOAD-TPR-PROGRAM                                                    |
| SYMBOL= <text 1..8>                                                   |
```

Das Kommando entlädt den gesamten Kontext mit diesem Symbol aus dem
System. Es wird nur mit dem Privileg TSOS akzeptiert. Fehlerhaft geladene
Routinen müssen ebenfalls mit diesem Kommando entladen werden.

Beschreibung der Operanden

SYMBOL
Entry oder Csect einer geladenen TPR-Routine

### 4.2.3 SHOW-TPR-PROGRAM-STATUS

```
| SHOW-TPR-PROGRAM-STATUS                                               |
|  SYMBOL=*ALL | *OWN | *SHARE | *SYSTEM                               |
|                                                                       |
| ,INFORMATION=*STD | *TASK                                             |
|                                                                       |
```

Das Kommando zeigt an, welche Entries aus welchen Bibliotheken geladen
sind und welchen Status sie haben.

INFORMATION=*STD
Symbol=*ALL Alle Symbole, die von Capri geladen wurden, werden angezeigt.
*OWN Nur die im eigenen Task(CL5) liegenden Symbole werden angezeigt.
*SHARE Alle Symbole, die in den CL4 share geladen wurden, werden angezeigt.
*SYSTEM Alle direkt aufgerufen Systemsymbole aus den gebundenen EXEC
werden angezeigt.

Die Ausgabe hat folgendes Format:

| TYP | SYMBOL | ADDRESS | STATE | RC | LIBRARY |
|---|---|---|---|---|---|
| C01 | TSTROUT | EC000080 | LOADED | 00000000 | $TSOS.SKMLNK.CAPRI.210 |
| C04 | TSTY | EF980CC0 | WARNING | 04010604 | $XXX.SKMLNK |
| E04 | TSTY1 | EF980D00 | WARNING | 04010604 | $XXX.SKMLNK |
| X | PBVSVS | CC4738C0 | SYSTEM | 00000000 | $TSOS.SYSTEM-ENTRY |
| C07 | ABCSECT | BEFF0010 | LOD_0AJK | 00000000 | $TSOS.SKMLNK.AB.010 |
| S06 | TEST99 | CF800000 | LDY_0ASJ | 00000000 | $TSOS.SKMLNK.TEST.011 |

| Spalte | Bedeutung |
|---|---|
| TYP=@nn | Typ des Symbols |
| | nn Nummer eines Link-Contexts, alle Symbole mit gleicher Nummer gehören zum selben Programm |
| | @ Symboltyp |
| | S Start/Csect |
| | E Entry |
| | I ILE |
| | C Common |
| | X System Entry des EXEC |
| SYMBOL | Name des geladenen Entry oder der Csect |
| ADDRESS | Adresse des Entry im CL2/CL4 oder CL5 des entsprechenden Tasks |
| STATE | LOADED Das Modul ist fehlerfrei geladen. |
| | WARNING Das Modul ist geladen; brachte aber Fehler. |
| | LOD_<tsn> Das Modul wurde tasklokal geladen. |
| | WRN_<tsn> Das Modul wurde tasklokal geladen, brachte aber Fehler. |
| | LDY_<tsn> Das Modul wurde von START-TPR-PROG taskspezifisch in den CL4-Speicher geladen. |
| | WDY_<tsn> Das Modul wurde von Start-TPR-PROG taskspezifisch in den CL4-Speicher geladen, brachte aber Warnings. |
| | SYSTEM Dieser Entry ist im gebundenen EXEC. |
| RC | Returncode vom $PBBND1 (Die Returncodes sind größtenteils mit den RC des BIND-Aufrufs identisch) |
| LIBRARY | Datei, aus der das Modul geladen wurde. |

INFORMATION=*TASK
Hier werden sämtliche Tasks vermerkt, deren Programm Counter sich zu diesem Zeitpunkt
gerade in der nachgeladenen TPR-Routine befinden.
Die Ausgabe hat folgendes Format:

| TSN | SYMBOL | CONTXT | | |
|---|---|---|---|---|
| 0ABC | TEST099 | $SYSTEM#CAPRI###1000004E | Context CL4 taskspezifisch | Suffix=TID |
| 0BCE | TEST05 | SYSTEM##CAPRI###20000050 | Context CL5 tasklokal | Suffix=TID |
| 0DEE | TEST002 | $SYSTEM#CAPRI###00000004 | Context CL4 Systemglobal | Suffix=Counter |

```
TSN          Task Sequence Number
SYMBOL       Einsprungstelle der TPR-Routine
CONTEXT      Name des Bindecontexts
             $ = privilegierter Systemcontext
             Suffix = TID (Task ID) oder eindeutiger Zähler
```

## 4.3 Rückmeldungen

Alle Kommandos haben CMD-Returncodes.

Kommando erfolgreich abgearbeitet:

```
RC2:    X'00'
RC1:    X'00'
MRC:    C'CMD0001'
```

TPR-Programm liefert keinen Returncode:

```
RC2:    X'02'
RC1:    X'00'
MRC:    C'CAPNORC'
```

Fehler im Subsystem Capri:

```
RC2:    X'00'
RC1:    X'20'
MRC:    C'CAP@@@@'
```

Alle Mainreturncodes mit CAP.... sind in der Help-Datei vorhanden und können mit /HELP-MSG-INFO CAP.... abgefragt werden.

Fehler im TPR-Programm:

```
RC2:    X'00'
RC1:    X'40'
MRC:    C'@@@@@@@'
```

Der Mainreturncode stammt aus dem TPR-Programm und wird im Feld CAPKEY übergeben.

## 4.4 Redefinition des START-TPR-PROGRAM Kommandos in START-XXX.

Mit dieser Version ist es möglich für das START-TPR-PROGRAM ein eigenes START-XXX Kommando zu erzeugen.
Um dies zu ermöglichen, sind im Programm einige Vorkehrungen zu treffen. Die wichtigste ist, die internen Operandennamen so zu ändern, dass sie systemweit eindeutig sind und Capri sie trotzdem erkennt.
Der interne Operandenname ist normalerweise $CAPXexe. Die Überprüfung erfolgt ab Stelle 5 auf "EXE". Das $ steht für eine privilegierte Schnittstelle.
Die Zeichenfolge "CAP" sollte durch das 3stellige Komponentenkennzeichen und das "X" durch ein frei wählbares Zeichen ersetzt werden.
Im nächsten Kapitel befindet sich ein Template des /START-xxx Kommandos.

## 4.5 /START-XXX Template

Das Template ist für die IMON-Schnittstelle

```
       //ADD-COMMAND NAME=START-<xxx>,INTERNAL-NAME=$<ccc><f>EXE,                -
             STANDARD-NAME=START-<xxx>,                                          -
             ALIAS-NAME=<xxx>,                                                   -
             DOMAIN=(SYSTEM-MANAGEMENT,PROGRAM),                                 -
             IMPLEMENTOR=P2(ENTRY=NLKRES56,INTERFACE=ISL,                        -
                 CMD-INTERFACE=*TRANSFER-AREA(MAX-STRUC-OPERAND=11)),            -
             PRIVILEGE=(STD-PROC)

       //ADD-OPERAND NAME=FROM-FILE,INTERNAL-NAME=FROFI                          -
             STANDARD-NAME=FROM-FILE,RESULT-OPERAND-NAME=*POSITION(5),           -
             DEFAULT='*BY-IMON'
       //ADD-VALUE KEYWORD(STAR=MAN),STRUCTURE=Y,VALUE='*BY-IMON'

       //ADD-OPERAND NAME=LOGICAL-ID,INTERNAL-NAME=ID,STANDARD-NAME=*NAME,       -
             DEFAULT=C'<syslnk>'(ANALYSE-DEFAULT=NO)
       //ADD-VALUE TYPE=COMP-NAME(LONG=30,SHORT=1)

       //ADD-OPERAND NAME=INSTALLATION-UNIT,INTERNAL-NAME=UNA,DEFAULT='<unit>',  -
-
             STANDARD-NAME=INSTALLATION-UNIT,RESULT-OPERAND-NAME=*POS(POS=7)
       //ADD-VALUE TYPE=COMP-NAME(LONG=30,SHORT=1)

       //ADD-OPERAND NAME=VERSION,INTERNAL-NAME=UVER,STANDARD-NAME=VERSION,      -
             RESULT-OPERAND-NAME=*POS(POS=8),DEFAULT='<*std>'
       //ADD-VALUE TYPE=PRODUCT-VERSION(USER-INT=*ANY(CORR-STATE=*ANY)
       //ADD-VALUE TYPE=KEYWORD(STAR=MAN),VALUE='*STD'

       //ADD-OPERAND NAME=SYMBOL,INTERNAL-NAME=ENTR,STANDARD-NAME=SYMBOL,        -
             DEFAULT=<symbol>,RESULT-OPERAND-NAME=*POS(POS=1)
       //ADD-VALUE TYPE=COMPOSED-NAME(LONG=8,SHORT=1,UNDERSCORE=YES)

       //ADD-OPERAND NAME=DEFAULT-LIBRARY,INTERNAL-NAME=DLIB,DEFAULT='<filename>',-
             STANDARD-NAME=DEFAULT-LIBRARY,RESULT-OPERAND-NAME=*POS(POS=9)
       //ADD-VALUE TYPE=FILENAME
       //CLOSE-STRUCTURE
```

Ersetzbare Werte:

| | |
|---|---|
| <xxx> | Programmname |
| <ccc> | 3-stelliges Komponentenkennzeichen |
| <f> | frei wählbares Zeichen<br>cccf muss systemweit eindeutig sein |
| <syslnk> | logical ID der Ladebibliothek |
| <unit> | Installation Unit siehe IMON |
| <*std> | neuste Version (*STD) oder Versionsnummer |
| <symbol> | Einsprungstelle des TPR-Moduls |
| <filename> | Dateiname der Ladebibliothek falls über IMON nicht ermittelbar |

# 5 Programmschnittstelle

## 5.1 SVC79

### 5.1.1 Alte kompatible Schnittstelle

Der SVC79 in der heutigen Form hat folgende Funktion: Das Register 1
wird mit der Adresse der TPR-Routine geladen und anschließend wird der
SVC abgesetzt. Ist der User privilegiert, so wird diese Routine im TPR-
Mode angesprungen. Ist keine Privilegierung vorhanden, so kehrt der SVC
ohne Ausführung des TPR-Codes zum Aufrufer zurück und setzt das Programm
direkt nach dem Aufruf fort.

### 5.1.2 Neue Schnittstelle

Im Register 1 der alten Schnittstelle ist immer eine Befehlsadresse, d.h., die
Adresse muss immer gerade sein. Dieser Umstand wird ausgenutzt, um die neue SS
zu kennzeichnen. Ist das 2**0-Bit gesetzt, so ist dies die neue Schnittstelle.
Die neue Schnittstelle besitzt eine Parameterleiste, in der der Entry Name
vermerkt ist. Das Subsystem CAPRI weist dem Entry eine Adresse zu und
springt diese Routine an. Die Parameterleiste wird im Register 1 übergeben.
Ist CAPRI nicht geladen, so darf diese Schnittstelle nicht aufgerufen werden.
Es kommt zu einem Systemdump im Modul ETMSF oder in der TPR-Routine mit IW 5C.

## 5.2 Beschreibung

### 5.2.1 Direkte Ausführung

Die TPR-Routine muss dem Subsystem in einer Bibliothek zur Verfügung
gestellt werden. Dieses Modul muss entsprechend der Systemkonvention
übersetzt sein.

### 5.2.2 Alter SVC79

Der alte SVC79 kann in der heutigen Form aufgerufen werden, wobei die
TPR-Routine ein Teil des Programmes ist.
Es gibt jetzt aber auch die Möglichkeit, den TPR-Teil mittels
Systemverwalterkommandos vorzuladen. Die Adresse des TPR-Entry kann
ein Programm, das unter TSOS läuft, mittels der Binder/Lader-Schnittstelle
VSVI1 ermitteln. Das Register 1 wird mit dieser Adresse der TPR-Routine versorgt
und diese Adresse wird über den SVC79 angesprungen.
Die alte Schnittstellenspezifikation bleibt erhalten.

### 5.2.3 Neuer SVC79

Der neue SVC79 hat eine Parameterleiste (Kap. 5.4.2); in dieser steht der
Name des Entry. Ist der Entry vorgeladen, so wird dieser angesprungen.
Ist der Entry nicht vorgeladen, so kann durch die Versorgung des
Bibliotheksnamen der Entry automatisch von CAPRI nachgeladen werden.

Das Modul kann anhand des Parameters MEMCL in den CL4- oder CL5-Speicher geladen werden. Im CL4-Speicher muss diese Routine für den normalen Betrieb reentrant sein.

Der Parameter PERM steuert die Dauer vom Laden bis zum Entladen des Moduls. In Verbindung mit CL4 verhält sich der Aufruf wie ein explizites LOAD-TPR-PROGRAM Kommando. Das Modul wird nicht mehr entladen.

Ist der CL5 Parameter gesetzt, so wird das Modul bei Programm-Terminierung entladen.

Mit dem Parameter FUNCALL=Y können Systemroutinen direkt angesprungen werden. Im Parameter ENTRY wird der System Entry oder ein *SVCnnn angegeben, mit nnn = SVC-Nummer. Im Parameter USERAREA muss dann die Parameterleiste der Systemfunktion übergeben werden. Die Parameterleiste wird dem System in Register 1 direkt übergeben und die entsprechende Funktion wird ausgeführt. Das Userprogramm wird nach dem Aufruf fortgesetzt.

## 5.3 Sicherheit

### 5.3.1 Class2 Option SVC79

Die Ausführung der CAPRI-Funktionen lässt sich wie beim SVC79 über die CLASS2-Option SVC79 steuern.

| | |
|---|---|
| SVC79=3 | Das Subsystem CAPRI lässt sich nicht laden. |
| SVC79=1 | Das Kommando START-TPR-PROGRAM oder der CAPINT-Aufruf muss von der Konsole bestätigt werden (MSG CAP0007). |
| SVC79=0 | Das Kommando START-TPR-PROGRAM oder der CAPINT-Aufruf werden sofort ausgeführt. |

Das Subsystem ist so ausgelegt, dass die Ladebibliothek xxxLNK.CAPRI.010 sowie die TPR-Modul-Bibliotheken auf user-access=*owner-only (share=no) stehen können. Der Systemverwalter ist verantwortlich, dass nur von ihm ausgewählte TPR-Routinen in dieser Bibliothek abgelegt werden.

### 5.3.2 GUARD

Der Systemverwalter (Userid TSOS) kann unabhängig von der CL2-Option SVC79, diese Funktion auch anderen Userids zur Verfügung stellen. Diese Freigabe wird mittels einer Coownerschaft auf die Ladebibliothek bewirkt. Die Funktion der Coownerschaft ist im SECOS-Manual Band 2 beschrieben. Die im Folgenden beschriebenen Kommandos sind nötig, um einem oder mehreren Usern einen Zugriff auf ein TPR-Modul zu ermöglichen. Sie sind nur ein Teil der Möglichkeiten, die die GUARDs bieten.
Es wird empfohlen, nur Ladebibliotheken freizugeben, die nur im Zugriff des Systemverwalters sind. Diese Bibliotheken sollten user-access=*owner-only (share=no) erklärt werden.

#### 5.3.2.1 Einrichten einer ACCESS CONDITION

Als erstes muss eine Verbindung zwischen der Userid und den Zugriffsrechten erstellt werden. Hierfür ist das Kommando:

```
     ADD-ACCESS-CONDITION
         GUARD-NAME=<Dateiname des protection guard>
         ,SUBJECTS=*USER(USER-ID=<name 1..8> list-poss(20))
         ,ADMISSION=*YES
```

Der <Dateiname des protection guard> ist die Verbindung zwischen dem Access-GUARD und dem Coowner-GUARD. Die einfachste ADMISSION ist: *YES, der Zugriff ist erlaubt. (Weitere Möglichkeiten stehen im SECOS-Manual Band 2)

#### 5.3.2.2 Einrichten der SVC79-Coownerschaft

Jetzt muss die Verbindung zwischen dem zu ladenden Modul und dem Access GUARD hergestellt werden. Dies geschieht mit dem Kommando:

```
     ADD-COOWNER-PROTECTION-RULE
         RULE-CONTAINER-GUARD=SYS.UCC
        ,PROTECTION-RULE=<Verwaltungsname>
        ,PROTECTION-OBJECT=
              *PARAMETER(NAME=<composed-name>
                              ,CONDITION-GUARD=<Dateiname des protection guard>)
```

Der <Verwaltungsname> ist ein Ordnungsbegriff innerhalb des RULE-CONTAINERS. Der <composed-name> stellt die Verbindung zwischen dem Modul und dem Schutz dar. Dieser Name ist aufgebaut wie ein Dateiname, bei dem auch Wildcards zugelassen sind. CAPRI baut aus der Userid, dem Bibliotheksnamen und dem zu ladenden Entry einen neuen Dateinamen auf. Dieser hat folgendes Format:

<userid>.<libname 1..24>.<entry>

Endet der auf 24 Zeichen abgeschnittene Dateiname mit Punkt, so wird dieser als Trennung verwendet.
Es gibt für die Systemaufrufe feste Dateinamen:

```
System Entry:    TSOS.SYSTEM-ENTRY.<entry>
SVC:             TSOS.SYSTEM-SVC.SVC<nnn>
```

Beispiel:

Eine Bibliothek mit Namen SYSLNK.HUGO.030 steht unter $TSOS und aus ihr wird der Entry HUGOSTRT geladen.

CAPRI erstellt folgenden zu überprüfenden Dateinamen:

$TSOS.TSOS.SYSLNK.HUGO.030.HUGOSTRT

Der <composed-name> im GUARD muss dann

TSOS.SYSLNK.HUGO.030.HUGOSTRT sein.

Wenn alle Symbole aus dieser Bibliothek zugreifbar sein sollen, ist auch folgender Name möglich:

TSOS.SYSLNK.HUGO.030.*

Steht diese Bibliothek unter der Kennung $ABC, so ist der zu überprüfende Name:

$TSOS.ABC.SYSLNK.HUGO.030.HUGOSTRT

## 5.4 Aufruf

### 5.4.1 Alter SVC79

Beim SVC 79 ist die Registerversorgung entsprechend der Systemfunktion $FNAT.

Register:

| | |
|---|---|
| R1 | Startadresse der Routine |
| R2 | A(TCB) |
| R3 | A(XVT) |
| R4 | A(unterbrochener PCB, Auslöser des Aufrufs) |
| R5 | PC des unterbrochenen PCB |

Alle anderen Register haben die Werte der aufrufenden Routine.

### 5.4.2 Neuer SVC79

Der neue SVC79 wird über den Makro CAPINT aufgerufen.

| Operation | Operanden |
|---|---|
| CAPINT | ENTRY=name |
| | |
| | ,LIBNAM=pathname |
| | [,PAR=*NONE \| text 1..64] |
| | [,DIAG=N\|Y] |
| | [,PERM=N\|Y] |
| | [,MEMCL=4\|5] |
| | [,MSG=N\|Y] |
| | [,FUNCALL=N\|Y] |
| | [,USERAREA=adr] |
| | MF=(E,adr) \| (E,(reg)) \| L \| C \| (C,pre) \| D \| (D,pre) |

ENTRY — Name des zu ladenden Symbols, System Entries oder *SVCnnn mit nnn = SVC-Nummer

LIBNAM — Vollständiger Pfadname einer Bibliothek, in der die Routine mit dem Entry liegt. Diese Bibliothek braucht nicht SHARE=YES zu sein.

PAR — Dieses ist ein 64 Byte langes Feld, das der Routine übergeben wird. Es dient zur Parametrisierung der TPR-Routine, falls gewünscht.

```
DIAG        Dieser Parameter ist für den Test der TPR-Routine gedacht.
            Bei "Y" wird die Verarbeitung nach den Laden angehalten.

PERM        Dieser Parameter ermöglicht das permanente Laden der Routine
            Bei "N" wird das Modul nach jedem Aufruf entladen.
            Bei "Y" erfolgt das Entladen entsprechend der Speicherklasse.

               CL4: Das Entladen der Routine kann nur mit Kommando
                    UNLOAD-TPR-PROGRAM erfolgen.
               CL5: Bei der Programm-Termination wird das Modul entladen.

MEMCL       Dieser Parameter gibt die Speicherklasse an, in der das Modul
            geladen wird. Bei MEMCL=5 wird das Modul taskspezifisch
            geladen; dadurch können nicht reentrant Routinen geladen werden.

MSG         "N" Dieser Parameter unterdrückt die Ausgabe von MSG
            aus dem Subsystem CAPRI auf SYSOUT.

FUNCALL     Dieser Parameter ermöglicht den direkten Ansprung einer
            Systemroutine oder eines System-SVCs. Bei "Y" wird von CAPRI
            mittels $VSVI1 der Entry Name im System gesucht. Die
            Parameter für diese Routine müssen in der USERAREA liegen.
            CAPRI springt anschließend diese Routine direkt an.


USERAREA    Dieser Parameter ermöglicht der TU-Routine, der TPR-Routine
            einen Datenbereich zum Datentransfer oder bei einem direkten
            Systemaufruf bzw. System-SVC die Aufrufparameter zu übergeben.
```

## 5.5 Interface Area (CAPINT)

```
CAPINT      DSECT
CAPHDR      DS      0F              Header
CAPFCTU     DS      H'480'          Function unit
CAPFCT      DS      X'00'           Function
CAPFCTV     DS      X'01'           Function Version Interface Version 01
CAPRC       DS      F               Return code
CAPDEF      EQU     X'FFFFFFFF'     RC undefined initial value
CAPOK       EQU     X'00000000'     RC Okay
            DS      C
CAPKEY      DS      CL7             MSGKEY as Insert for MSG CAP0003
CAPENTR     DS      CL8             Name of called entry
CAPLIB      DS      CL54            Name of Library
CAPPAR      DS      CL64            Parameter like CMD-Parameter
CAPNONE     EQU     C'*NONE'        No parameter available
CAPUADR     DS      A               Address of user area
CAPIND      DS      X               Indicator
CAPPERM     EQU     X'01'           Routine is loading permanently
CAPDIAG     EQU     X'02'           Stop after loading
CAPNOMS     EQU     X'04'           no Message to SYSOUT
CAPMC5      EQU     X'10'           Memory class 5 for bind (task local)
CAPSYCA     EQU     X'20'           Entry is system entry or SVC
CAPICALL    DS      CL3             Indicator of calling interface
CAPCMD      EQU     C'CMD'          Interface is called by START-TPR-PRO
CAPSVC      EQU     C'SVC'          Interface is called by SVC
CAPEADR     DS      A               Address of Entry
CAPTCB      DS      A               Address of TCB
```

```
CAPEXVT         DS       A              Address of EXVT
CAPISTK         DS       A              Address of interrupted PCB
CAPINIA         DS       A              Address of predecessor PCB
CAPSVR1         DS       F              Register 1 after SVC return
CAPSVR15        DS       F              Register 15 after SVC return
                DS       6F             reserved for future use
CAPEND          DS       0F             End of interface area
CAPLEN          EQU      *-CAPINT       Length of parameter
```

Parameterbeschreibung:

| | |
|---|---|
| CAPFCTV | Interfaceversionsnummer V01 |
| CAPRC | Returncode F'00' Routine korrekt ausgeführt<br>CAPKEY=CAPOKAY, sonst siehe MF=D |
| CAPKEY | Dieser Schlüssel wird bei der Endemeldung ausgegeben. Beginnt der Schlüssel mit CAP, so ist dieser RC von CAPRI und kann mit /HELP-MSG-INFO CAPaaaa analysiert werden. |
| CAPENTR | Name des aufgerufenen Symbols, eines System Entry oder eines SVCs |
| CAPPAR | 64 Byte langer Übergabeparameter. Dieser Parameter kann auch mit dem Kommando START-TPR-PROGRAM übergeben werden. |
| CAPLIB | Dateiname der Ladebibliothek |
| CAPUADR | Adresse eines Userbereichs. Diese Adresse wird von CAPRI ungeprüft weitergegeben. Bei System Entry oder SVC muss hier die Adresse der Parameterleiste stehen. |
| CAPPERM | Die Routine wird nur einmal geladen und dann nur noch angesprungen. |
| CAPDIAG | Es wird nach dem Laden und vor dem Entladen eine beantwortbare Meldung ausgegeben. Sie ist sinnvoll bei der Diagnose der TPR-Routine. |
| CAPNOMS | Es werden alle Ausgaben unterdrückt und nur Returncodes übergeben. |
| CAPMC5 | Tasklokales Laden der Routine (sie muss hierfür nicht reentrant sein); nur in Verbindung mit CAPPERM=Y sinnvoll. |
| CAPSYCA | Das in CAPENTR angegebene Symbol ist ein System Entry oder ein System-SVC. |
| CAPEADR | Adresse des Symbols (nur für Diagnosezwecke; nicht für direkten Ansprung geeignet). |
| CAPSVR1 | Register 1 nach Ausführung des SVCs, z.B. nötig bei $REQM |
| CAPSVR15 | Register 15 nach Ausführung des SVCs, z.B. nötig bei $REQM |

## 5.6 Programmierregeln

### 5.6.1 Neuer SVC79

Der Aufruf erfolgt folgendermaßen:

```
                        CAPINT  MF=(E,CAPPAR)
                        ...
                        .....
         CAPPAR         CAPINT  MF=L,LIBNAM=LIBABC,ENTRY=TPRENTRY
```

Die TPR-Routine folgt den DSL-Konventionen.

```
        <entry>         @ENTR   TYP=E,ENV=SPLSPEC,RETURNS=NO
                        USING   CAPINT,R1
```

```
                        ......
                        MVC     CAPRC,...
                        MVC     CAPKEY,=CL7'_______'
                        @EXIT
                        @END
```

Ansprung einer Systemroutine

```
                        LA      3,VSVI
                        USING   VSVIDS,R3
                        LA      4,SYSCALL
                        USING   CAPINT,4
                        ......
                        CAPINT  MF=(E,SYSCALL)
                        @IF     EQ
                        CLC     CAPRC,=F'00'   Prüfen, ob SVC79 okay
                        @THEN
                        @IF     EQ
                        CLC     PBVSVRET,=F'00' Prüfen, ob Systemaufruf okay
                        @THEN
                        ...
                        ....
                        @END
SYSCALL                 CAPINT  ENTRY=PBVSVS,FUNCALL=Y, -
                                USERAREA=VSVI, MF=L
VSVI                    $VSVI1  MF=L,SELECT=ALLLIST;SCOPE=GLOBAL, -
                                INCTX=RCTX,OUTADDR=INS1,OUTLEN=10000, -
                                RUNMOD=ADV
CAPINT                  CAPINT  MF=D
VSVIDS                  $VSVI1  MF=D
                        ....
```

Ausführung eines SVC

```
                        LA      4,SYSSVC
                        USING   CAPINT,4
                        ......
                        CAPINT  MF=(E,SYSSVC)
                        @IF     EQ
                        CLC     CAPRC,=F'00'    Prüfen, ob SVC79 okay
                        @THEN
                        @IF     EQ
                        CLC     CAPSVR15,=F'00' Hier das R15 nach
                                                dem SVC, sonst im Parameter
                        @THEN
                        L       R1,CAPSVR1      Hier das R1 nach dem SVC
                        ...
                        ....
                        @END
SYSSVC                  CAPINT  ENTRY=*SVC228,FUNCALL=Y, -
                                USERAREA=REQM,MF=L
REQM                    $REQM   MF=L,CLASS=5,BYTES=X'2000'
CAPINT                  CAPINT  MF=D
                        ....
```

### 5.6.2 Alter SVC79

Die Routine muss nach folgendem Schema programmiert sein:

```
<TPR-modul>    CSECT
               ENTRY    <entry>
<entry>        @ENTR    TYP=D
               LR       R10,R1    Register 10 ist das Basisregister der Routine
               USING    <entry>,R10
               [USING   ETCB,R2]  Register 2 zeigt auf den TCB
               [USING   EXVT,R3]  Register 3 zeigt auf die EXVT
               [USING   ESTK,R4]  Register 4 zeigt auf den interrupted PCB
* R13 muss auf eine Savearea von 18F und R12 auf die Adresse V(ITSPRV) zeigen
*************************************************************

               Hier ist die eigentliche Routine


               $FNDT       Rücksprung zum Aufrufer
               @END
               DTCB     MF=D
               DXVT     MF=D
               DSTK     MF=D
               END
```

## 5.7 Compileroptionen

Für die Übersetzung der privilegierten Routine sind folgende
ASSEMBLER/ASSTRAN-Parameter nötig:

### 5.7.1 /390

ASSEMBH oder ASSTRAN

MODULE-GENERATION(MODE=STD[,DESTINATION-CODE=STD])

### 5.7.2 x86

ASSTRAN ab 5.2B

MODULE-GENERATION(MODE=STD,DESTINATION-CODE=X86(RUN-MODE=TPR))