Deutsch

Fujitsu Software BS2000

COSMOS

Benutzerhandbuch

Stand der Beschreibung:
COSMOS V21.0

Ausgabe Juni 2021

Kritik… Anregungen… Korrekturen…

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an bs2000.info@fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2015

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001: 2015 erfüllt.

Copyright und Handelsmarken

Copyright © 2025 Fujitsu

Alle Rechte vorbehalten.
Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

FUJITSU

COSMOS

Continuous Monitoring System

Sequence Number		
Volume	**46**	Automated System Operations
Chapter	**52**	External Interface Specification
Section	**31**	COSMOS
Ident	**\210**	COSMOS V21.0A
Version	**001**	2021-06-09
Author		COSMOS-Team

Copyright © 2025 Fujitsu Germany GmbH
All rights reserved

Table of contents

1 Changes in COSMOS

1.1 Changes for BS2000 V21.0A (COSMOS V21.0A)

1. No changes.

1.2 Changes for BS2000/OSD-BC V11.0A (COSMOS V20.0A)

1. No changes.

1.3 Changes for BS2000/OSD-BC V10.0A (COSMOS V19.0A)

1. No changes.

1.4 Changes for BS2000/OSD-BC V9.0A (COSMOS V18.0A)

1. The maximum value of the BUFFER-SIZE parameter in the SET-COSMOS-PARAMETERS instruction is increased from 7 to 40.
2. The maximum value of the NUMBER-OF-BUFFERS parameter in the SET-COSMOS-PARAMETERS instruction is increased from 128 to 512.
3. In the EVENT-SELECTION parameter of the SET-COSMOS-PARAMETERS instruction the CPU-NUMBER can be specified to select events according to selected CPU numbers.

1.5 Changes for BS2000/OSD-BC V8.0A (COSMOS V17.0A)

1. No changes.

1.6 Changes for BS2000/OSD-BC V7.0A (COSMOS V16.0A)

1. No changes.

1.7 Changes for BS2000/OSD-BC V6.0B (COSMOS V15.0B)

1. New events RSCS and RSCT.

1.8 Changes for BS2000/OSD-BC V6.0A (COSMOS V15.0A)

1. New events SWSR and DSM.

1.9 Changes for BS2000/OSD-BC V5.0A (COSMOS V14.0A)

1. No changes.

1.10 Changes for BS2000/OSD-BC V4.0A (COSMOS V13.0A)

1. New events TICS and TICE (collective name TIC).
2. New events KCOL and KPST (collective name KAI).
3. New event HAL.
4. New parameter MEASUREMENT-TIME for instruction SET-COSMOS-PARAMETERS.
5. For the selection operands TSN, CATEGORY, JOB-NAME and USER-ID of the parameter TASK-SELECTION of instruction SET-COSMOS-PARAMETERS an asterisk is permitted as the last character in the wildcard definition.
6. For the selection operand TSN of the parameter TASK-SELECTION of instruction SET-COSMOS-PARAMETERS up to 16 TSNs can be specified.
7. Monitoring of PPAT no longer supported

1.11 Changes for BS2000/OSD-BC V3.1B (COSMOS V12.1B)

1. New design for event PTSK (#12).

2. The emulated /390-time is output in the events STAT (#8), DEST (#36), PEND (#52), UPND (#112).
3. New data MODE = RISC | /390 is output in event TSKI (#48).
4. The name of the virtual machine on which BS2000/OSD is running is output in event INIT (#4).

2 Introduction

This document describes the measurement tool COSMOS. The main aim of COSMOS is to help individuals to evaluate the various aspects of system behaviour of interest to them.

This document is primarily written for those individuals who are not familiar with COSMOS and what it does, but would like to know something about it or perhaps may intend to use it.

COSMOS is an internal product delivered as a subsystem. The subsystems BCAM-COS and DCAM-COS are necessary if DCM events are opened.

COSMOS is not a static tool and the course its development takes and its degree of usefulness are largely dependent on the needs and suggestions of its users.

3 COSMOS Overview

COSMOS is a software monitoring system that runs as an extension of the BS2000 operating system and captures a trace of pertinent system data on a file, which is subsequently post processed and printed. COSMOS runs in processor state TPR (under a userid with the privilege SW-MONITOR-ADMINISTRATION) to establish the trace tape control, process the parameters and establish the user/system interface. The actual data capturing portions of COSMOS run either in state TPR or state SIH as appropriate, as individual extensions of the pertinent system modules where the desired data is generated or processed.

The gathering code is loaded from the COSMOS SYSLNK by SM2 using DSSM ($ESMCRE SSNAME=COSMOS). After the measurement the COSMOS modules can be unloaded by SM2 ($ESMDEL SSNAME=COSMOS).

The system commands /START-SUBSYSTEM, /DELETE-SUBSYSTEM, /HOLD-SUBSYSTEM and /RESUME-SUBSYSTEM are not supported for COSMOS.

It is possible to write data either to a PAM-file or to tape. The rate of missed events using PAM will be much higher than using tape. In order to reduce the number of missed events, one should specify a file command with a space-parameter before executing COSMOS.

Example:

/CREATE-FILE FILE-NAME=...,SUPPORT=PUB-DISK(SPACE=REL(1000,100))

The file should be big enough in order to avoid a secondary allocation. If the secondary allocation is set to 0 by using the parameter SPACE=REL(xxxx,0) of the /CREATE-FILE command the user can limit the size of the disk file. In normal write mode the COSMOS write task gets a DMS error and is disabled when the size xxxx is reached (→ abnormal termination of COSMOS !). In wrap-around mode the file is overwritten periodically. To reduce the rate of missed events in case of output to disk file, the use of a dedicated private disk is recommended. To reduce the rate of missed events in case of output to tape, it is possible to write data in streaming mode.

Each discrete data group logged on tape or disk is called a COSMOS event and is given a four character name. At chosen points in the system where interesting data is generated or changed, standard COSMOS *hooks* are created; each *hook* being associated with a particular event name (ref. chapter Table of Events). Depending on user parameters, the event is *opened* i.e. enabled allowing COSMOS event data gathering code to become active at each pass through the system module hook. When COSMOS is terminated, all currently open system *hooks* are closed i.e. disabled.

COSMOS has the ability to gather data selectively on the basis of specified tasks. Some COSMOS events are mandatory i.e. they are always recorded on tape and therefore need not be *opened* by the user. The rate at which data is generated depends on how busy the system is and how many events are currently *opened*. To give a very approximate idea of tape usage, a heavily loaded system with all events opened will fill a tape reel in less than 10 minutes.

COSMOS handles multi-volume tape files. If information is being generated at such a rate that a tape volume is being filled in an unacceptably short time, then probably COSMOS is not being used judiciously (i.e. too many events are open) and furthermore, the effect of COSMOS itself on the system performance would be significant. A software monitor, by its very nature, must influence that which it is trying to measure, but the aim of each measurer should be to reduce this influence to a minimum. Where possible, measurement sessions should be organised so that they are oriented towards recording as few events as possible during one particular session. Sessions can then be repeated with different events opened. This is preferable than having one session with many events opened, although not always possible in practice.

COSMOS maintains and outputs to the COSMOS file an internal counter containing the number of missed events. However, there is no indication of which particular events were missed.

The accumulation of missed events works as follows : each buffer output by COSMOS contains a counter of missed events which gives the number of events missed between the

last event recorded in the previous output buffer and the first event recorded in this buffer. NEGET passes this information to the user (see NEGET description).

Included automatically in any COSMOS run is a user supplied run title, the system modules address map EOLDTAB, data concerning the configuration, the date and time of the run, the available physical core for the session together with the mandatory event information.

4 Statements for COSMOS

The individual COSMOS statements (SM2 syntax) are described in the following sections.

4.1 Statements for Defining the Monitoring Routine

For each monitoring routine to be started, the objects (events) to be monitored and the parameters to be monitored must be specified. For this purpose ADD, REMOVE, SET and MODIFY statements are provided.

The ADD statement defines precisely the events to be measured in the monitoring program. The REMOVE statement can be used to exclude a set of monitored events defined with the aid of ADD statements (the mandatory events can't be removed).

The SET statement defines a list of parameters for monitoring. The MODIFY statement can be used to modify the values of the measurement program already defined with the SET statement. Actual monitoring of defined measurement program does not commence until the monitoring program run is started.

4.1.1 ADD-COSMOS-EVENT

FUNCTION:

The ADD statement defines precisely the events to be measured in the monitoring program.

FORMAT:

ADD-COSMOS-EVENT

EVENT-NAME = list-poss(70):*STANDARD-EVENTS

*ACF	*BCAM	*BCPT	*BLS	*BOUR	*CHTM	*CMD
*CMS	*DAB	*DCAM	*DLM	*DSM	*EIA2	*EIA3
*FITC	*GSAC	*HAL	*IDLE	*INTR	*IONQ	*ISEV
*ISPL	*KAI	*LOCK	*MSG	*NSM	*P2PM	*PAGE
*PAM	*PCCC	*PCTC	*PDEA	*PEND	*PIO	*PMIO
*PRGS	*PRGT	*PRTY	*RELM	*REQM	*RSCS	*RSCT
*SDV	*SLOT	*SNAP	*STD1	*STDI	*STDN	*SVC
*SWSR	*TGMA	*TGMP	*TGMT	*TIC	*TINF	*TLM
*TLT	*TSKI	*TSVC	*UTM	*VMCH	*VMH	*VMI
*VMLK	*VMPD	*VMPR	*VMS	*VM2	*WSCT	*XEIA

DESCRIPTION OF THE OPERANDS:

EVENT-NAME =

This operand defines which events, others than the mandatory events, are to be opened during the run.

EVENT-NAME = STANDARD-EVENTS

opens following events: ACF, BLS, BOUR, CHTM, CMS, DAB, EIA2, EIA3, FITC, IDLE, INTR, IONQ, PAGE, PAM, PCCC, PCTC, PEND, PMIO, PRGS, PRGT, RELM, REQM, SDV, SVC, TSKI, TSVC, WSCT.

EVENT-NAME

*ACF	|*BCAM	|*BCPT	|*BLS	|*BOUR	|*CHTM	|*CMD
*CMS	|*DAB	|*DCAM	|*DLM	|*DSM	|*EIA2	|*EIA3
*FITC	|*GSAC	|*HAL	|*IDLE	|*INTR	|*IONQ	|*ISEV
*ISPL	|*KAI	|*LOCK	|*MSG	|*NSM	|*P2PM	|*PAGE
*PAM	|*PCCC	|*PCTC	|*PDEA	|*PEND	|*PIO	|*PMIO
*PRGS	|*PRGT	|*PRTY	|*RELM	|*REQM	|*RSCS	|*RSCT
*SDV	|*SLOT	|*SNAP	|*STD1	|*STDI	|*STDN	|*SVC
*SWSR	|*TGMA	|*TGMP	|*TGMT	|*TIC	|*TINF	|*TLM
*TLT	|*TSKI	|*TSVC	|*UTM	|*VMCH	|*VMH	|*VMI
*VMLK	|*VMPD	|*VMPR	|*VMS	|*VM2	|*WSCT	|*XEIA

The events corresponding to the names of the list-elements will be opened. The event P2PM is not displayed in guided dialog.

4.1.2 REMOVE-COSMOS-EVENT

FUNCTION:

The REMOVE statement can be used to exclude events defined with the aid of ADD statement.

FORMAT:

REMOVE-COSMOS-EVENT

DESCRIPTION OF THE OPERANDS:

same operands as ADD-COSMOS-EVENT
additional the value *ALL for the operand EVENT-NAME

EVENT-NAME = *ALL

All events except the mandatory events PTSK, CREA, DEST, INIT, STAT, MMRC and LGON will be removed.

4.1.3 SET-COSMOS-PARAMETERS

FUNCTION:

This statement defines the conditions of the monitoring program. The mandatory events (PTSK, CREA, DEST, LGON, MMRC, INIT, STAT) are always opened and couldn't be closed with the statement REMOVE. The files, which are specified in the operands OUTPUT and VM2000, must be defined previously.

FORMAT

SET-COSMOS-PARAMETERS
TITLE = C'COSMOS' | <c-string 1..80>
,NUMBER-OF-SLOTS = 8 | <integer 1..20>
,BUFFER-SIZE = 7 | <integer 1..40>
,NUMBER-OF-BUFFERS = 32 | <integer 2..512>
,ADDITIONAL-INFO=*NONE | list-poss(2): *CONFIGURATION |*VM2000(...)
*VM2000 (...)
FILE-NAME = <full-filename 1..54>
,OUTPUT = *DISK(...) | *WRAP-AROUND (...) | *TAPE (...) | *STREAM-TAPE (...)

```
        *DISK (...)
        FILE-NAME = list-poss (16) : <full-filename 1..54>
        *WRAP-AROUND (...)
        FILE-NAME = <full-filename 1..54>
        *TAPE (...)
        FILE-NAME = list-poss (16) : <full-filename 1..54>
        *STREAM-TAPE (...)
        FILE-NAME = list-poss (16) : <full-filename 1..54>
    ,TASK-SELECTION= *SPECIFIED(...) | *ALL
        *SPECIFIED(...)
            JOB-NAME = *NOT-SPECIFIED | list-poss(8):<alphanum-name 1..8 with-wild>
            ,CATEGORY = *NOT-SPECIFIED | list-poss(8):<alphanum-name 1..7 with-wild>
            ,USER-ID = *NOT-SPECIFIED | list-poss(8):<alphanum-name 1..8 with-wild>
            ,TSN = *NOT-SPECIFIED | list-poss(16):<alphanum-name 1..4 with-wild>
            ,TYPE = *NOT-SPECIFIED | list-poss(4): *SYSTEM | *BATCH | *DIALOG | *TP
    ,EVENT-SELECTION = *SPECIFIED(...) | *ALL-BY-ADD-COSMOS-EVENT
        *SPECIFIED(...)
            EIA-INTERRUPT-CLASS = *ANY | list-poss(5) : *S / *P / *M / *I / *E
            ,EIA-SVC-NUMBER = *ANY | list-poss(8) : <integer 1..255>
            ,IO-DEVICE = *ANY | list-poss(8) : <alphanum-name 2..4>
            ,DAB-CACHE-ID = *ANY | list-poss(8) : <alphanum-name 1..32>
            ,MEMORY-CLASS = *ANY | list-poss(4) : 3 | 4 | 5 | 6
            ,SLOT-MEMORY-CLASS = *ANY | list-poss(4) : 3 | 4 | 5 | 6
            ,PEND-CODE = *ANY | list-poss(16) : <integer 1..22>
            ,LOCK-ID = *ANY | list-poss(4) : <alphanum-name 1..2>
            ,TLT-DESCRIPTOR = *ANY | list-poss(8) : <alphanum-name 1..3>
            ,TSKI-SWITCH = *ANY | *TASK
            ,TSVC-SVC-NUMBER = *ANY | list-poss(8) : <integer 1..255>
            ,CPU-NUMBER = *ANY | list-poss(32) : <integer 0..31>
    ,UNLOAD = *AT-MEASUREMENT-PROGRAM-STOP | *AT-SM2-STOP
    ,MEASUREMENT-TIME = *NOT-SPECIFIED | <integer 1..60>
```

DESCRIPTION OF THE OPERANDS:

TITLE =

TITLE = C'COSMOS'

When no title is specified by the user of COSMOS, the default COSMOS will be assumed.

TITLE = <c-string 1..80>

This is an up to eighty byte free form run title used to identify the run. This title is copied into the first data block written to the tape from where it can be extracted by the appropriate reduction program. The evaluation program CAP for example prints this title at the top of every page.

NUMBER-OF-SLOTS =

NUMBER-OF-SLOTS = <integer 1..20>

Specifies the number of slots for interruptible events. The parameter is not displayed in guided dialog.

NUMBER-OF-BUFFERS =

NUMBER-OF-BUFFERS = 32 | <integer 2..512>

To reduce the rate of missed events, it is possible to specify the number of COSMOS buffers.

The buffers are requested dynamically (CLASS 3 memory).

In case this operand is not supplied, a default value of 32 is assumed.

If the value specified for this operand is less than (2 * # output files), it is set to (2 * # output files) resp. to (4 * # output files) in case of streaming mode.

BUFFER-SIZE =

BUFFER-SIZE = 7 | <integer 1..40>

To reduce the rate of missed events, it is possible to specify the size (in 4K-PAGES) of the COSMOS buffers.

In case this operand is not supplied, a default value of 7 is assumed.

ADDITIONAL-INFO =

This operand signals COSMOS that at begin of the measurement some additional information should be written to COSMOS file and specifies the file for the VM2000-Events.

ADDITIONAL-INFO = *NONE
No additional information should be written to COSMOS file.

ADDITIONAL-INFO = list-poss (3): *CONFIGURATION | *VM2000 (...)

*CONFIGURATION

The configuration is written to COSMOS file.

*VM2000(...):
This operand defines the file for VM2000-Hypervisor events.

FILE-NAME = <full-filename 1..54>
The VM2000 events are output to a separate file. The file must be created before the measurement is started. The other /CREATE-FILE parameters are the same as for COSFIL.

OUTPUT =

This operand defines, in which mode the data is written.

OUTPUT = *DISK(...)

In case this operand is not supplied, the default value *DISK is assumed.

FILE-NAME = list-poss (16): <full-filename 1..54>
It specifies the COSMOS files to which the event records are written in parallel by different task. Each file must be created by a /CREATE-FILE command before starting the measurement. For each file a buffer-write-task is created. In case of a DMS open error for one of this files COSMOS issues a warning message and terminates. After the measurement the files must be mixed together to one complete file before evaluations with CAP or NEGET are started (see chapter COSMIX – COSMOS Output File Mix Program, page 31).

OUTPUT = *WRAP-AROUND(...)

FILE-NAME = <full-filename 1..54>
It specifies a COSMOS file which is overwritten cyclically. The file must be created by a /CREATE-FILE command before starting the measurement. The secondary allocation of the file must be set to zero.

OUTPUT = *TAPE (...)

FILE-NAME = list-poss (16): <full-filename 1..54>
It specifies the COSMOS files to which the event records are written in parallel by different tasks. Each file must be created by a /CREATE-FILE command before starting the measurement.

OUTPUT = *STREAM-TAPE (...)

FILE-NAME = list-poss (16): <full-filename 1..54>
It specifies the COSMOS files to which the event records are written in parallel by different tasks. The COSMOS data are written in streaming mode. Each file must be created by a /CREATE-FILE command before starting the measurement.

TASK-SELECTION =

This operand is used to specify the tasks to be monitored. Task dependent data will be gathered.

TASK-SELECTION = *SPECIFIED

The tasks to be monitored are selected via their TSN, USER-ID, JOB-NAME, CATEGORY or their TYPE.

JOB-NAME =

JOB-NAME = *NOT-SPECIFIED
The selection is not depending on the job name.

JOB-NAME = list-poss(8): <alphanum-name 1..8 with-wild >

This operand selects the individual job names for which task dependent data will be monitored.
An asterisk ("*") is permitted as the last character in the wildcard definition.
A warning occurs in case of non-existent job names!

CATEGORY =

CATEGORY = *NOT-SPECIFIED
The selection is not depending on the category name.

CATEGORY = list-poss(8): <alphanum-name 1..7 with-wild >
This operand selects the individual categories for which task dependent data will be gathered.
An asterisk ("*") is permitted as the last character in the wildcard definition.
A warning occurs in case of non-existent category names!

USER-ID =

USER-ID = *NOT-SPECIFIED
The selection is not depending on the USER-ID.

USER-ID = list-poss(8): <alphanum-name 1..8 with-wild>
This operand selects the USER-IDs for which task dependent data will be monitored.
An asterisk ("*") is permitted as the last character in the wildcard definition.
A warning occurs in case of non-existent USER-IDs!

TSN =

TSN = *NOT-SPECIFIED
The selection is not depending on the TSN.

TSN = list-poss(16): <alphanum-name 1..4 with-wild>
This operand selects the individual TSN's for which task dependent data will be gathered.
An asterisk ("*") is permitted as the last character in the wildcard definition.
An asterisk ("*") is not permitted for TSNs with leading blanks (e.g. " PT5", " TM").
A warning occurs in case of non-existent TSNs!

TYPE =

This operand selects the class or classes for which task dependent data will be gathered.

Dynamic changes of the task type (category) will not become known to COSMOS ! e.g. a BATCH task changed to SYSTEM will still be tracked if only BATCH tasks should be tracked; a BATCH task changed to TP will not be tracked even if TP tasks are tracked.

TYPE = *NOT-SPECIFIED
The selection is not depending on the task attribute.

TYPE = list-poss(4): *SYSTEM | *BATCH | *DIALOG | *TP
*SYSTEM: all system tasks are to be monitored .
*BATCH : all batch tasks are to be monitored.
*DIALOG: all dialog tasks are to be monitored.
*TP : all TP tasks are to be monitored.

Note:

The selection operands TYPE, TSN, CATEGORY, JOB-NAME and USER-ID and are also applied to new created tasks (during create resp. logon).

TASK-SELECTION = *ALL
All tasks are monitored without restrictions.

EVENT-SELECTION =

This operand is used to specify restrictions of events to be monitored.

EVENT-SELECTION = *ALL-BY-ADD-COSMOS-EVENT
No restrictions are specified.

EVENT-SELECTION = *SPECIFIED
The recording of events is only done, if the specified restrictions come true.

EIA-INTERRUPT-CLASS =

EIA-INTERRUPT-CLASS = list-poss(16): <alphanum-name 1..1>
EIA events are selected, using the interruption codes.
EIA-INTERRUPT-CLASS = *ANY
No interruption code restrictions are specified.

EIA-SVC-NUMBER =

EIA-SVC-NUMBER = list-poss(8): <integer 1..255>
EIA events are selected using the SVC number.
EIA-SVC-NUMBER = *ANY
No EIA events are selected via SVC's.

IO-DEVICE =

The events CHTM, IONQ, SDV and PMIO are selected, using the device mnemonic.
IO-DEVICE = list-poss(8): <alphanum-name 2..4>
The events CHTM, IONQ, SDV and PMIO are monitored for the specified device mnemonics.
IO-DEVICE = *ANY
No restrictions concerning the device mnemonics are specified. All events (CHTM, IONQ, SDV and PMIO) will be gathered.

DAB-CACHE-ID =

The events DAB are selected, using the cache id.
DAB-CACHE-ID = list-poss(8): <alphanum-name 1..32>
The events DAB are monitored for the specified cache ids.
DAB-CACHE-ID = *ANY
No restrictions are specified. All DAB events will be gathered.

MEMORY-CLASS =

MEMORY-CLASS = list-poss(4): 3 | 4 | 5 | 6
Only the events corresponding to the specified memory class are gathered.
MEMORY-CLASS = *ANY
No memory class restrictions are specified. All RELM and REQM events are gathered.

SLOT-MEMORY-CLASS =

SLOT-MEMORY-CLASS = list-poss(8): 3 | 4 | 5 | 6
Only the SLOT events corresponding to the specified memory class are gathered.
SLOT-MEMORY-CLASS = *ANY
No memory class restrictions are specified. All SLOT events are gathered.

PEND-CODE =

PEND-CODE = list-poss(16): <integer 1..22>
Only the events PEND and UPND, corresponding to the specified pend codes, will be gathered.
PEND-CODE = ANY
No pend code restrictions are specified. All PEND and UPND events are gathered.

LOCK-ID =

LOCK-ID = list-poss(4): <alphanum-name 2..2>
Only the events corresponding to the specified LOCK-ID are gathered.
LOCK-ID = *ANY
No LOCK-ID restrictions are specified. All LOCK events will be gathered.

TLT-DESCRIPTOR =

TLT-DESCRIPTOR = list-poss(8): <alphanum-name 3..3>
Only the events corresponding to the specified TLT description (combination ENTRY-FUNCTION-SUCCES) will be gathered.
For the meaning of ENTRY-FCT-SUCCESS in the event TLT see the corresponding description in NEGET.
TLT-DESCRIPTOR = *ANY
No restrictions are specified. All TLT events will be gathered.

TSKI-SWITCH =

TSKI-SWITCH = *TASK
The contents of the event TSKI is reduced.

TSKI-SWITCH = *ANY
No restriction is specified. The complete TSKI event is gathered.

TSVC-SVC-NUMBER =

TSVC-SVC-NUMBER = list-poss(8): <integer 1..255>
Only the events corresponding to the specified SVC# are gathered.
TSVC-SVC-NUMBER = *ANY
No restrictions are specified. All TSVC events will be gathered.

CPU-NUMBER =

CPU-NUMBER = list-poss(32): <integer 0..31>
Only the events occurring on the specified CPU are gathered.
CPU-NUMBER = *ANY
No restrictions are specified. All events will be gathered.

UNLOAD =

UNLOAD = *AT-MEASUREMENT-PROGRAM-STOP
The COSMOS-code is unloaded at the termination of the measurement program.
UNLOAD = *AT-SM2-STOP
The subsystem COSMOS is unloaded at the termination of the software monitor SM2

MEASUREMENT-TIME =

MEASUREMENT-TIME = *NOT-SPECIFIED
The measurement must be stopped explicitly by the instruction //STOP-MEASUREMENT-PROGRAM TYPE=COSMOS

MEASUREMENT-TIME = <integer 1..60>
The measurement will be stopped automatically after the specified number of minutes.

NOTES :

- Additionally the selected events have to be opened with aid of statement ADD-COSMOS-EVENT.
- In case of task-dependent events, the COSMOS event handler checks at first whether the concerning task should be tracked, then the selection is done.

4.1.4 MODIFY-COSMOS-PARAMETER

FUNCTION:
This statement defines the modifications of the SM2 settings for the monitoring program COSMOS.

FORMAT:

MODIFY-COSMOS-PARAMETER
TITEL = *UNCHANGED | <c-string 1..80>
,NUMBER-OF-SLOTS = *UNCHANGED | 8 | <integer 1..20>
,BUFFER-SIZE = *UNCHANGED | <integer 1..40>
,NUMBER-OF-BUFFERS = *UNCHANGED | <integer 2..512>
,ADDITIONAL-INFO = *UNCHANGED| *NONE | list-poss (3): *CONFIGURATION | *VM2000 (...)
*VM2000 (...)
FILE-NAME = <full-filename 1..54>
,OUTPUT = *UNCHANGED| *DISK(...) | *WRAP-AROUND (...) |*TAPE(...) | *STREAM-TAPE(...)
*DISK(...)
FILE-NAME = list-poss (16): <full-filename 1..54>
*WRAP-AROUND (...)
FILE-NAME = <full-filename 1..54>
*TAPE(...)
FILE-NAME = list-poss (16): <full-filename 1..54>
*STREAM-TAPE(...)
FILE-NAME = list-poss (16): <full-filename 1..54>
,TASK-SELECTION= *UNCHANGED | *ALL | *SPECIFIED(...)
*SPECIFIED(...)
JOB-NAME = *NOT-SPECIFIED | list-poss(8) : <alphanum-name 1..8 with-wild>
,CATEGORY = *NOT-SPECIFIED | list-poss(8): <alphanum-name 1..7 with-wild>
,USER-ID = *NOT-SPECIFIED | list-poss(8) : <alphanum-name 1..8 with-wild>

```
        ,TSN = *NOT-SPECIFIED | list-poss(16) : <alphanum-name 1..4 with-wild>
        ,TYPE = *NOT-SPECIFIED | list-poss(4): *SYSTEM | *BATCH | *DIALOG | *TP
  ,EVENT-SELECTION = *UNCHANGED | *ALL-BY-ADD-COSMOS-EVENT | *SPECIFIED(...)
    *SPECIFIED(...)
      EIA-INTERRUPT-CLASS = *UNCHANGED | *ANY | list-poss(16) : <alphanum-name 1..1>
      ,EIA-SVC-NUMBER= *UNCHANGED | *ANY | list-poss(8):<integer 0..255>
      ,IO-DEVICE = *UNCHANGED | *ANY | list-poss(8) : <alphanum-name 2..4>
      ,DAB-CACHE-ID = *UNCHANGED | *ANY | list-poss(8) : <alphanum-name 1..32>
      ,MEMORY-CLASS = *UNCHANGED | *ANY | list-poss(4) : 3 | 4 | 5 | 6
      ,SLOT-MEMORY-CLASS = *UNCHANGED | *ANY | list-poss(4) : 3 | 4 | 5 | 6
      ,PEND-CODE =*UNCHANGED | *ANY | list-poss(16):<integer 1..22>
      ,LOCK-ID = *UNCHANGED | *ANY | list-poss(4) : <alphanum-name 1..2>
      ,TLT-DESCRIPTOR = *UNCHANGED | *ANY | list-poss(8) : <alphanum-name 1..3>
      ,TSKI-SWITCH = *UNCHANGED | *ANY | *TASK
      ,TSVC-SVC-NUMBER = *UNCHANGED | *ANY | list-poss(8):<integer 0..255>
      ,CPU-NUMBER = *UNCHANGED | *ANY | list-poss(32) : <integer 0..31>
  ,UNLOAD = *UNCHANGED | *AT-MEASUREMENT-PROGRAM-STOP | *AT-SM2-STOP
  ,MEASUREMENT-TIME = *UNCHANGED | *NOT-SPECIFIED | <integer 1..60>
```

DESCRIPTION OF THE OPERANDS:

TITLE =

TITLE = *UNCHANGED
The currently defined title remain unchanged.

TITLE = <c-string 1..80>
This is an up to eighty byte free run title used to identify the run. This title is copied into the first data block written to the tape, from where it can be extracted by the appropriate reduction program. The evaluation program CAP for example prints this title at the top of every page.

NUMBER-OF-SLOTS =

NUMBER-OF-SLOTS = *UNCHANGED
The currently defined number of slots remain unchanged.

NUMBER-OF-SLOTS = <integer 1..20>
Specifies the number of slots for interruptible events. The parameter is not displayed in guided dialog.

NUMBER-OF-BUFFERS =

NUMBER-OF-BUFFERS = *UNCHANGED
The currently defined number of buffers remain unchanged.

NUMBER-OF-BUFFERS = <integer 2..512>
To reduce the rate of missed events, it is possible to specify the number of COSMOS buffers.
The buffers are requested dynamically (CLASS 3 memory).

BUFFER-SIZE =

BUFFER-SIZE = *UNCHANGED
The currently defined number of buffer pages remain unchanged.
BUFFER-SIZE = <integer 1..40>
To reduce the rate of missed events, it is possible to specify the size (in 4K-pages) of the COSMOS buffers.

ADDITIONAL-INFO =

This operand signals COSMOS, that at begin of the measurement some additional information should be written to COSMOS file and specifies the file for the VM2000 events.

ADDITIONAL-INFO = *UNCHANGED
The currently defined additional information remain unchanged.

ADDITIONAL-INFO = *NONE
No additional information should be written to COSMOS file.

ADDITIONAL-INFO = list-poss(2): *CONFIGURATION | *VM2000(...)

CONFIGURATION:
The configuration is written to COSMOS file.

*VM2000(...):
This operand defines the file for VM2000-Hypervisor events.

FILE-NAME = <full-filename 1..54>
The VM2000 events are output to a separate file. The file must be created before the measurement is started. The other /CREATE-FILE parameters are the same as for COSFIL.

OUTPUT =
This operand defines, in which mode COSMOS writes the data in a file.
In case this operand is not supplied, the default value *DISK is assumed.

OUTPUT = *UNCHANGED
The currently defined output mode remain unchanged.

OUTPUT = *DISK(...)
FILE-NAME = list-poss (16): <full-filename 1..54>
It specifies the COSMOS files to which the event records are written in parallel by different task. Each file must be created by a /CREATE-FILE command before starting the measurement.
For each file a buffer-write-task is created. In case of a DMS open error for one of this files COSMOS issues a warning message and terminates.
After the measurement the files must be mixed together to one complete file before evaluations with CAP or NEGET are started (see chapter COSMIX – COSMOS Output File Mix Program, page 31).

OUTPUT = *WRAP-AROUND(...)
FILE-NAME = <full-filename 1..54>
It specifies a COSMOS file which is overwritten cyclically.
The file must be created by a /CREATE-FILE command before starting the measurement. The secondary allocation of the file must be set to zero.

OUTPUT = *TAPE (...)
FILE-NAME = list-poss (16): <full-filename 1..54>
It specifies the COSMOS files to which the event records are written in parallel by different tasks. Each file must be created by a /CREATE-FILE command before starting the measurement.

OUTPUT = *STREAM-TAPE (...)
FILE-NAME = list-poss (16): <full-filename 1..54>
It specifies the COSMOS files to which the event records are written in parallel by different tasks. The COSMOS data are written in streaming mode. Each file must be created by a /CREATE-FILE command before starting the measurement.

TASK-SELECTION =
This operand is used to specify the tasks to be monitored.
Task dependent data will be gathered.

TASK-SELECTION = *UNCHANGED
The currently defined task selection remain unchanged.

TASK-SELECTION = *SPECIFIED
The tasks to be monitored are selected via their TSN, USER-ID, JOB-NAME, CATEGORY or their TYPE.

JOB-NAME =
JOB-NAME = *NOT-SPECIFIED
The selection is not depending on the job name.

JOB-NAME = list-poss(8): <alphanum-name 1..8 with-wild >
This operand selects the individual job names for which task dependent data will be monitored.
An asterisk ("*") is permitted as the last character in the wildcard definition.
A warning occurs in case of non-existent job names !

CATEGORY =
CATEGORY = *NOT-SPECIFIED
The selection is not depending on the category name.

CATEGORY = list-poss(8): <alphanum-name 1..7 with-wild >

This operand selects the individual categories for which task dependent data will be gathered.
An asterisk ("*") is permitted as the last character in the wildcard definition.
A warning occurs in case of non-existent category names!

USER-ID =

USER-ID = *NOT-SPECIFIED
The selection is not depending on the USER-ID.

USER-ID = list-poss(8): <alphanum-name 1..8 with-wild >
This operand selects the USER-IDs for which task dependent data will be monitored.
An asterisk ("*") is permitted as the last character in the wildcard definition.
A warning occurs in case of non-existent USER-IDs!

TSN =

TSN = *NOT-SPECIFIED
The selection is not depending on the TSN

TSN = list-poss(16): <alphanum-name 1..4 with-wild>
This operand selects the individual TSNs for which task dependent data will be gathered.
An asterisk ("*") is permitted as the last character in the wildcard definition.
An asterisk ("*") is not permitted for TSNs with leading blanks (e.g. " PT5", " TM").
A warning occurs in case of non-existent TSNs!

TYPE =

This operand selects the class or classes for which task dependent data will be gathered. Dynamic changes of the task type (category) will not become known to COSMOS ! e.g. a BATCH task changed to SYSTEM will still be tracked if only BATCH tasks should be tracked; a BATCH task changed to TP will not be tracked even if TP tasks are tracked.

TYPE = *NOT-SPECIFIED
The selection is not depending on the task attribute.

TYPE = list-poss(4): *SYSTEM | *BATCH | *DIALOG | *TP
*SYSTEM: all system tasks are to be monitored.
*BATCH : all batch tasks are to be monitored.
*DIALOG: all dialog tasks are to be monitored.
*TP : all TP tasks are to be monitored.

NOTE: The selection operands TYPE, TSN, CATEGORY, JOB-NAME and USER-ID and are also applied to new created tasks (during create resp. logon).

TASK-SELECTION = *ALL
All tasks are monitored without restrictions.

EVENT-SELECTION =
This operand is used to specify restrictions of events to be monitored.

EVENT-SELECTION = *UNCHANGED
The currently defined event selection remain unchanged.

EVENT-SELECTION = *ALL-BY-ADD-COSMOS-EVENT
No restrictions are specified.

EVENT-SELECTION = *SPECIFIED
The recording of events is only done if the specified restrictions come true.

EIA-INTERRUPT-CLASS =
The EIA events are selected according to the interrupt class.
EIA-INTERRUPT-CLASS = *UNCHANGED
The currently defined interrupt class restrictions remain unchanged.

EIA-INTERRUPT-CLASS = list-poss(16): <alphanum-name 1..1>
EIA events are selected using the interrupt class.

EIA-INTERRUPT-CLASS = *ANY
No interrupt class restrictions are specified.

EIA-SVC-NUMBER =
The EIA events are selected according to the SVC number.

EIA-SVC-NUMBER = *UNCHANGED
The currently defined SVC number restrictions remain unchanged.

EIA-SVC-NUMBER = list-poss(8): <integer 1..255>
EIA events are selected using the SVC number.

EIA-SVC-NUMBER = *ANY
No SVC number restrictions are specified.

IO-DEVICE =
The IONQ, SDV, CHTM and PMIO events are selected using the device mnemonic.

IO-DEVICE = *UNCHANGED
The currently defined device mnemonic restrictions remain unchanged.

IO-DEVICE = list-poss(8): <alphanum-name 2..4>
The IONQ, SDV, CHTM and PMIO events are only monitored for the specified device mnemonics.

IO-DEVICE = *ANY
No device mnemonic restrictions are specified. All IONQ, SDV, CHTM and PMIO events will be gathered.

DAB-CACHE-ID =
The DAB events are selected using the cache id.

DAB-CACHE-ID = *UNCHANGED
The currently defined cache id restrictions for DAB events remain unchanged.

DAB-CACHE-ID = list-poss(8): <alphanum-name 1..32>
The DAB events are monitored only for the specified cache ids.

DAB-CACHE-ID = *ANY
No restrictions are specified. All DAB events will be gathered.

MEMORY-CLASS =
MEMORY-CLASS = *UNCHANGED
The currently defined memory class restrictions for RELM and REQM events remain unchanged.

MEMORY-CLASS = list-poss(4): 3 | 4 | 5 | 6
Only the events corresponding to the specified memory class are gathered.

MEMORY-CLASS = *ANY
No memory class restrictions are specified. All RELM and REQM events are gathered.

SLOT-MEMORY-CLASS =
SLOT-MEMORY-CLASS = *UNCHANGED
The currently defined memory class restrictions for SLOT events remain unchanged.

SLOT-MEMORY-CLASS = list-poss(8): 3 | 4 | 5 | 6
Only the SLOT events corresponding to the specified memory class are gathered.

SLOT-MEMORY-CLASS = *ANY
No slot memory class restrictions are specified. All SLOT events are gathered.

PEND-CODE =
PEND-CODE = *UNCHANGED
The currently defined pend code restrictions for PEND and UPND events remain unchanged.

PEND-CODE = list-poss(16): <integer 1..22>
Only PEND and UPND events corresponding to the specified pend codes are gathered.

PEND-CODE = *ANY
No pend code restrictions are specified. All PEND and UPND events are gathered.

LOCK-ID =
LOCK-ID = *UNCHANGED
The currently defined lock id restrictions for LOCK events remain unchanged.

LOCK-ID = list-poss(4): <alphanum-name 2..2>
Only the events corresponding to the specified lock id are gathered.

LOCK-ID = *ANY
No lock id restrictions are specified. All LOCK events will be gathered.

TLT-DESCRIPTOR =
TLT-DESCRIPTOR = *UNCHANGED
The currently defined TLT descriptor remain unchanged.

TLT-DESCRIPTOR = list-poss(8): <alphanum-name 3..3>
Only the events corresponding to the specified TLT descriptor (combination ENTRY-FUNCTION-SUCCES) will be gathered. For the meaning of ENTRY-FCT-SUCCESS in the TLT event see the corresponding description in NEGET.

TLT-DESCRIPTOR = *ANY
No restrictions are specified. All TLT events will be gathered.

TSKI-SWITCH =
TSKI-SWITCH = *UNCHANGED
The currently defined TSKI switch remain unchanged.

TSKI-SWITCH = *TASK
The contents of the TSKI event is reduced.

TSKI-SWITCH = *ANY
No restriction is specified. The complete TSKI event is gathered.

TSVC-SVC-NUMBER =
TSVC-SVC-NUMBER = *UNCHANGED
The currently defined SVC number restrictions for TSVC events remain unchanged.

TSCV-SVC-NUMBER = list-poss(8): <integer 1..255>
Only the events corresponding to the specified SVC number are gathered.

TSVC-SVC-NUMBER = *ANY
No restrictions are specified. All TSVC events will be gathered.

CPU-NUMBER =
CPU-NUMBER = *UNCHANGED
The CPU numbers defined by this time will not be changed.
CPU-NUMBER = list-poss(32): <integer 0..31>
Only the events occurring on the specified CPUs are gathered.
CPU-NUMBER = *ANY
No restrictions are specified. All events will be gathered.

UNLOAD =
UNLOAD = *UNCHANGED
The type of unload defined by this time will not be changed.

UNLOAD = *AT-MEASUREMENT-PROGRAM-STOP
The subsystem COSMOS is unloaded at the termination of the measurement program.

UNLOAD = *AT-SM2-STOP.
The subsystem COSMOS is unloaded at the termination of the software monitor SM2

MEASUREMENT-TIME =
MEASUREMENT-TIME = *UNCHANGED
The measurement time defined by this time will not be changed.

MEASUREMENT-TIME = *NOT-SPECIFIED
The measurement must be stopped explicitly by the instruction //STOP-MEASUREMENT-PROGRAM TYPE=COSMOS

MEASUREMENT-TIME = <integer 1..60>
The measurement will be stopped automatically after the specified number of minutes.

NOTES :

- Additionally the selected events have to be opened with aid of statement ADD-COSMOS-EVENT.
- In case of task-dependent events, the COSMOS event handler checks at first whether the concerning task should be tracked, then the selection is done.
- Selection in DAB is only done for the subevents ANALYSIS and COMMAND.

4.2 Statements for Starting and Stopping the Monitor COSMOS

Following definition of events to be monitored, the monitoring program initialization and run can be started. After newly defining or modifying a measurement program, the COSMOS run must be stopped an then started again in order to actually add the newly defined events and parameters. To do so, INIT, START, CHANGE and STOP statements are used.

4.2.1 INITIATE-COSMOS

FUNCTION:

After this statement the measurement program COSMOS defined previously is completely prepared for monitoring but no events - not even mandatory events - will be gathered at this time.

FORMAT:

INITIATE-COSMOS

4.2.2 CHANGE-MEASUREMENT-PROGRAM

FUNCTION:
This statement is used to stop an active monitoring program run and to restart it using already defined monitored events or parameters.

FORMAT:
CHANGE-MEASUREMENT-PROGRAM
TYPE = COSMOS

DESCRIPTION OF THE OPERANDS:
TYPE = COSMOS
The monitoring program COSMOS is restarted using the modified parameters.

4.2.3 START-MEASUREMENT-PROGRAM

FUNCTION
This statement is used to start the run of the monitoring program specified by the TYPE operand.

FORMAT:
START-MEASUREMENT-PROGRAM
TYPE = COSMOS

DESCRIPTION OF THE OPERANDS:
TYPE = COSMOS
The monitoring program COSMOS, which must be defined previously, will be started.
The initialization of COSMOS is done implicitly if the monitoring program run wasn't prepared before with aid of the statement INITIATE-COSMOS.

4.2.4 STOP-MEASUREMENT-PROGRAM

FUNCTION:
This statement is used to terminate the monitoring program session for the monitoring program specified under TYPE

FORMAT:
STOP-MEASUREMENT-PROGRAM
TYPE = ... / COSMOS

DESCRIPTION OF THE OPERANDS:
TYPE = COSMOS
Terminates the monitoring program for COSMOS.

4.3 Statements for Measurement Information

4.3.1 SHOW-ACTIVE-PARAMETER

FUNCTION:
Information about the objects and parameters of the active monitoring program are issued.

FORMAT
SHOW-ACTIVE-PARAMETER
TYPE = ... / COSMOS

DESCRIPTION OF THE OPERANDS:
TYPE = COSMOS
Provide information about the active monitoring program COSMOS.

4.3.2 SHOW-DEFINED-PARAMETER

FUNCTION:

Information about the objects and parameters of the defined monitoring program specified with aid of TYPE are issued.

FORMAT

SHOW-DEFINED-PARAMETER
TYPE = ... / COSMOS

DESCRIPTION OF THE OPERANDS:

TYPE = COSMOS

Provide information about the monitoring program COSMOS defined at this time.
This statement could be used to check the parameters of the defined monitoring program COSMOS before start.

5 COSMOS Run

COSMOS can be run from a terminal or as a batch job.
The user must first LOGON under a userid with the privilege SWMONADM. Then a monitoring program analogous to the following one should be defined and executed.

1 /CREATE-FILE FILE-NAME=...,SUPPORT=TAPE(VOL=..., -
DEVICE-TYPE=..., PREMOUNT-LIST=0)

in case of a tape file or

2 /CREATE-FILE FILE-NAME=...,SUPPORT=PUBLIC-DISK(SPACE= -
REL(...,...))

in case of a disk file

3 /SHOW-TRACE-STATUS OUTPUT=*SYSLST (see below)

4 /START-SM2

5 //SET-COSMOS-PARAMETERS ....

6 //ADD-COSMOS-EVENT NAME = ...

7 //REMOVE-COSMOS-EVENT NAME=...

8 //INITIATE-COSMOS

9 //START-MEASUREMENT-PROGRAM TYPE=COSMOS

10 //MODIFY-COSMOS-PARAMETER ...

11 //CHANGE-MEASUREMENT-PROGRAM TYPE=COSMOS

12 //STOP-MEASUREMENT-PROGRAM TYPE=COSMOS

13 /SHOW-TRACE-STATUS OUTPUT=*SYSLST (see below)

The trace facilities of some components of BS2000-OSD have much influence on performance. To avoid misinterpretation of measured data it is necessary to know which traces have been active during measurement. The output of /SHOW-TRACE-STATUS should be given to the interpreter together with the files created by COSMOS.

6 COSMOS Macro NPFDSS for Start/Stop Event Recording

Event recording is controlled by the operands TASK-SELECTION and EVENT-SELECTION in the statement SET-COSMOS-PARAMETER.

There is additionally a COSMOS entry by which the recording of non-mandatory events can be stopped and started.

Interface :
assembler macro NPFDSS

Usage Restriction :
FL=TPR

Supported MF Values =
(D,E,L)

Syntax:
LABEL NPFDSS MF=D,PREFIX=<PREFIX>
LABEL NPFDSS MF=E,PARAM=<PARAM>
LABEL NPFDSS MF=L,FCT=<FCT>,PREFIX=<PREFIX>

Input Operands :
PREFIX : first letter of all generated names
default value : N

PARAM : address of parameter list
<PARAM> = <ADDR> | (<REG>)
<PARAM> = <ADDR> : <ADDR> specifies the name of the parameter list
<PARAM> = (<REG>): register <REG> contains address of parameter list

FCT : function
FCT = START : start data gathering
FCT = STOP : stop data gathering
default value : START

Parameter list :

```
LABEL    NPFDSS MF=D,PREFIX=D
LABEL    DSECT
DNPCSHDR FHDR MF=(D,D)
DNPCSRES DS A             RESERVED
DNPCSFCT DS F             CALLED FUNCTION
DNPCSFST EQU 1             START
DNPCSFSP EQU 0             STOP
DNPC#    EQU *-DNPCSHDR
```

Register Usage :
ISL-Linkage

Return Codes:
00800001 COSMOS not started
00400004 unknown function value

The entry may e.g. be called by test programs for recording of events only during special time intervals or if error conditions are detected. Previously COSMOS has to be started and the necessary events have also to be opened by COSMOS.

7 Table of Events

The following table contains the code, name and type of all events recorded by COSMOS.

Mandatory events (type = MAN) are always recorded and cannot be added (*opened*) or removed (*closed*).

Task-dependent events (type = TDE) are recorded only for tasks specified in the TASK-SELECTION operand of the SET-COSMOS-PARAMETERS instruction.

Not task-dependent events (type = NTD) are recorded independently of the specified task selection. The event names listed in the table below should be used to add or remove the corresponding events with the //ADD-COSMOS-EVENT resp. //REMOVE-COSMOS-EVENT instruction.

For some groups of events a summary name is provided to add resp. remove all events belonging to the group:

ADD-COSMOS-EVENT NAME=EIA	opens both events: EIA2 and EIA3
ADD-COSMOS-EVENT NAME=PEND	opens both events: PEND and UNPN.
ADD-COSMOS-EVENT NAME=DAB	opens all DAB-events.
ADD-COSMOS-EVENT NAME=DLM	opens all DLM-events.
ADD-COSMOS-EVENT NAME=NSM	opens all NSM-events.
ADD-COSMOS-EVENT NAME=VM2	opens all VM2000-Hypervisor-events
ADD-COSMOS-EVENT NAME=VMH	opens both events: VMHS and VMHE
ADD-COSMOS-EVENT NAME=VMI	opens both events: VMIS and VMIE
ADD-COSMOS-EVENT NAME=VMS	opens both events: VMSS and VMSE
ADD-COSMOS-EVENT NAME=TIC	opens both events: TICS and TICE
ADD-COSMOS-EVENT NAME=KAI	opens both events: KCOL and KPST

Code	Event	Type	Cause of Event
4	INIT	MAN	COSMOS initiation
8	STAT	MAN	COSMOS initiation + termination
12	PTSK	MAN	activation of periodic task PT5
16	SDV	TDE	start device instruction
20	EIA2	TDE	program independent interrupt occurred during SIH-state
24	IDLE	NTD	system idles
28	EIA3	TDE	interrupt occurred in program state TU or TPR
	(EIA3)		interrupts except SVCs
	(SVC)		only SVC interrupts
	(EIA)		to open all EIA events (incl. event 20 !)
	(XEIA)		extended EIA3 for SVC or program interrupt
32	CREA	MAN	creation of a task
36	DEST	MAN	destruction of a task
40	HAL	NTD	data from HAL
44	CHTM	TDE	channel termination
48	TSKI	TDE	initiation of a task
52	PEND	TDE	task (un)pended
56	ACF	NTD	activation control function
60	INTR	TDE	interval timer runout
64	RELM	TDE	release memory
68	REQM	TDE	request memory
72	RSCS	NTD	RSC-IO start
76	RSCT	NTD	RSC-IO termination
80	UTM	TDE	universal transaction monitor
84	BOUR	NTD	bourse call
92	PAGE	TDE	page fault
96	LOCK	NTD	nucleus lock request or release
100	IONQ	TDE	enqueue IO-slot
104	CMS	TDE	catalog access via CMS
116	PAM	TDE	IO via DQPAM
124	SLOT	TDE	use of dedicated slot pools
128	SWSR	NTD	system working set replacement
132	WSCT	NTD	change of the working set control
136	PRTY	TDE	internal priority change
140	TINF	TDE	change of task attributes
144	PDEA	TDE	page deactivation
148	TLT	NTD	call of a TLT-lock-function
156	SNAP	NTD	snapshot dump
160	DCAM	TDE	DCAM access method
164	BCAM	TDE	BCAM teleprocessing
168	BCPT	TDE	BCAM port service
172	TLM	TDE	call of the task lock manager
180	MSG	TDE	system message handling
184	PCCC	NTD	PCS category control
188	PCTC	TDE	PCS task control
192	CMD	TDE	SDF command handling
196	FITC	NTD	fast intertask communication
200	DAB	TDE	disc access buffer
204	PIO	NTD	paging IO
208	BLS	TDE	dynamic program loading/unloading
216	LGON	MAN	logon processing of a task
220	PRGS	NTD	program start
224	PRGT	NTD	program termination
228	ISPL	NTD	ISAM pool access
232	ISEV	NTD	ISAM internal block lock modification
236	TICS	TDE	task initiation
240	TICE	TDE	task deinitiation
244	STD1	NTD	user program event
248	STDI	NTD	standard event, interruptible

252	STDN	NTD	standard event, non-int.
256	DSM	TDE	data space manager
260	PMIO	TDE	filename for IO via DQPAM
264	KPST	NTD	KAI $POST request
268	KCOL	NTD	KAI $COLL request
280	TSVC	TDE	subfunctions of SVC's
284	VMHS	NTD	VM: start hypervisor mode
288	VMHE	NTD	VM: end hypervisor mode
292	DLM1	TDE	DLM is called for enqueue
296	DLM2	TDE	DLM is called for convert, dequeue or cancel
300	DLM3	TDE	DLM returns to caller after asynchronous call
308	TGMT	TDE	task joining or leaving a group
312	TGMA	TDE	assignment of a task group to CPU
316	TGMP	TDE	info. about the loads on the single processors
320	MMRC	MAN	main memory reconfiguration
324	DABI	TDE	DAB: Completion of data transfer request
328	DABC	TDE	DAB: Completion/Deletion of a cache buffer
332	DABF	TDE	DAB: Start/Stop of caching a file
336	DABE	TDE	DAB: Start/Stop of caching a volume extent
340	DABS	TDE	DAB: intermediate saving of cached data
344	NSMA	TDE	NSM: synchronous/asynchronous return to DLM
348	NSMB	TDE	NSM writes an lock request into the token container
352	NSMC	TDE	NSM sends the token to its successor node
356	NSMD	TDE	NSM: at token arrives at a specific node
360	NSME	TDE	NSM: a request is transferred to function unit lock server or is sent from lock server
364	DABA	TDE	DAB: Information on access to files
368	VMIS	NTD	VM: start hypervisor idle
372	VMIE	NTD	VM: end hypervisor idle
376	VMSS	NTD	VM: start scheduling
380	VMSE	NTD	VM: end scheduling
384	VMPD	NTD	VM: pend queue transition
388	VMLK	NTD	VM: lock
392	VMPR	NTD	VM: periodic routine
396	VMCH	NTD	VM: change (add, del, mod VM)
400	GSAC	TDE	Access to Global Storage

7.1 Standard Events (STDI, STDN)

New events are defined in co-ordination with COSMOS/CAP/NEGET. this means that it is not possible to introduce new COSMOS data records in a BS2000 version after its release.

The events STDI/STDN (# 248/252) provide an interface for system programs to write arbitrary (test) data records into COSMOS file. The event STDI has to be used for interruptible code, the event STDN for non-interruptible code.

The hooks are implemented in the normal way (see chapter COSMOS Interfaces, macro $NPFCHK). For event STDN (# 252) the COSMOS call must be implemented as $NPFCHK TYP=CALL,STATE=NN and register 11 must be loaded with the address of the EXVT.

In both events register 1 points to a user data record of the following format :

- 1 byte length field : length of the following data
- user data

If the events are opened COSMOS copies this record into COSMOS file and inserts additional data: timestamp, event#, LM#, return address to the hook and the TSN of the task in control (only for STDI).

The user data should start with an identification (printable) of the caller (user) to have a distinction between different users of the event.

7.2 User Program Events (STD1)

The event STD1 (# 244) provides an interface for user programs to write data records into COSMOS file.

Two values of the call parameter list define the data to be output:

- start address of data
- length of data

COSMOS inserts additional data into the event record : timestamp, event#, LM#, TSN of the user task, address of COSMOS call.

The user data must start with a 4 char identification (printable) defined by the user ! This is necessary for evaluators to select special events.

It is clear that data recording is only done after COSMOS start and if the event STD1 is open. The COSMOS call (hook) is realized by a SVC. To avoid the SVC overhead in case of closed event STD1 the user should implement an extra switch for the COSMOS call.

The hooks are implemented via macro NPCHK (see chapter COSMOS Interfaces).

8 COSMIX – COSMOS Output File Mix Program

If the COSMOS data of one measurement are written into more than one output file (COSMOS parameter FILE-NAME) then this files must be mixed together to one measuring file before evaluations are started. This is done by the program COSMIX.

Input files :

COSMOS files of the measurement with the link names COSFIL, COSFIL02, ..., COSFILn
For the first file specified in the FILE-NAME-Parameter the link name COSFIL has to be used.

Output file :

New file with link name COSMIX (rest of the /FILE params are the same as for COSFIL)
COSMIX reads the following parameters from SYSDTA :

1 BLOCKS=N (optional)
N (decimal number) data blocks of the COSMOS files are copied into the output file.
Maximum value is 999999999.
COSFIL is copied at least up to the first event block.

In case this parameter is omitted, the whole data of the measurement are copied.

2 END (mandatory)
End of parameters. The copy program starts now.

9 Appendix - Notes and Restrictions

1 The details about the information and data layout of COSMOS events may be found in the NEGET description. Anyone wishing to write a data reduction program for COSMOS events should use this NEGET interface.

2 The task parameters are cumulative. Tracking of a task cannot be disabled until the end of the measurement.

3 If many events are opened and the number and size of COSMOS buffers are not increased (via parameters BUFFER= and BUFFER-PAGES=) then a lot of events will be lost.

4 If DCM specific events are to be measured, the subsystems BCAM-COS and DCAM-COS are necessary. Subsystem UTM must be started if UTM events are opened.

5 Leading blanks of parameter values must be given (e.g. in parameter TSN=).

6 In the operational-mode is the COSMOS-trace for nucleus locks only with the utility NLMSERVE switchable.

7 In the streaming mode COSMOS can't write about tape end.

8 VM2000-Hypervisor-events can be gathered on a VM2000-Monitor-System only.

9 HAL-events can be gathered on SR2000- or SX-Systems only.

10 COSMOS Interfaces

This chapter describes assembler macros and SPL4 include members to insert COSMOS measuring points in system modules and user programs.

10.1 NPCHK - Macro to insert COSMOS Hooks in User Programs

The macro NPCHK defines COSMOS measuring points in user programs written in assembler, the parameter list for the call and the corresponding DSECT.

User Description

Usage Restriction :
FL=TU

Supported MF Values =
(D,E,L)

Syntax:
[LABEL] NPCHK MF=D [,PREFIX=<PREFIX>]
[LABEL] NPCHK MF=E [,PARAM=<PARAM>]
[LABEL] NPCHK MF=L [,DATAPT=<ADDR> ,DATALN=<VALUE>]

Input Operands :
PREFIX : first letter of all generated names
default value : C
PARAM : address of parameter list
<PARAM> = <ADDR> | (<REG>)
<PARAM> = <ADDR> : <ADDR> specifies the name of the parameter list
<PARAM> = (<REG>): register <REG> contains address of parameter list
DATAPT : pointer (label) to user data for output
DATALN : length of data (bytes)
max. value : 240
default value : 0

Parameter list :
LABEL NPCHK MF=D
LABEL DSECT
CNPCHDR FHDR MF=(C,CNPC)
CNPCPTR DS A DATA POINTER
CNPCLNG DS Y DATA LENGTH
CNPC# EQU *-CNPCHDR

Call Interface :
SVC 12

Return Codes:
00400004 EVENT STD1 NOT OPEN
00400010 VALIDATION ERROR FOR PARAMLIST (WRITE) OR USER DATA (READ)
00400014 INVALID DATA LENGTH

The return codes are stored in REG.15 and in the header (field CNPCRET), except RC=00400010 in case of validation error for parameter list (RC only in REG.15).

The user data record is written to the COSMOS file if the event STD1 is open (see also chapter User Program Events).

Example

```
NPCHKDS  NPCHK MF=D                          SVC PARAMS DSECT
                  ...
         <coding>
         ...
         LA       R1,NPCHKPL                 SVC PARAMS
         USING NPCHKDS,R1
         MVC      SVCDDD(L'KOHL),KOHL        DATA FOR OUTPUT
         LA       R2,4+L'KOHL                TOTAL LENGTH
         STH      R2,CNPCLNG                 INTO PARAMS
         NPCHKMF=E,PARAM=(R1)                CALL COSMOS
         ...
         <data>
         ...
NPCHKPL  NPCHKMF=L,DATAPT=SVCDATA            SVC PARAMS
*
SVCDATA  DS       0A                         DATA FOR OUTPUT :
SVCDID   DC       CL4'0815'                  IDENTIFICATION
SVCDDD   DS       XL256                      REST OF DATA
*
KOHL     DC    ....
```

10.2 $NPFCHK - Macro to insert COSMOS Hooks in System Modules

The macro $NPFCHK serves to introduce COSMOS measuring points (so called *hooks*) in system modules written in assembler to connect them to the measuring monitor COSMOS.

A COSMOS measuring point consists of three parts. These will start with 3 invocations of the macro $NPFCHK.

The first call (TYP=BEGIN) defines the measuring point (event name, event number, subevent name, subevent number, ...) and generates code to switch the measuring point on or off.

If the measuring point is switched off, the program will continue at the physical end of the measuring point (TYP=END, third call). In the other case the program continues with the next statement after the first $NPFCHK call.

The historical way to switch a measuring point on or off generates a branch statement at the first call of $NPFCHK, which at switch-on time will be transformed by COSMOS to a NOP. For events which require high performance this logic can (demanded by a parameter) still be used. For events without these high requirements or where the measuring points are located in dynamically loaded components, a bit set by COSMOS in the system will be tested and if it is not set the program will continue after the physical end of the measuring point.

Afterwards data for the call of the COSMOS routine (the so-called event handler) corresponding to this measuring point can be gathered.

The call of this COSMOS routine will be done by the macro $NPFCHK TYP=CALL.

After the call the old module environment (as before the first $NPFCHK call) can be restored.

The third call (TYP=END) ends the measuring point.

Dynamically these 3 calls always have to be in the sequence defined above (TYP=BEGIN, CALL, END).

Physically TYP=END has to be given after TYP=BEGIN. It is recommended to use the sequence (BEGIN, CALL, END) always. Meanwhile it is possible to place the TYP=CALL outside of this sequence.

A COSMOS hook consists of the sequence of calls :

```
$NPFCHK TYP=BEGIN,...
<identification of the measuring point>
        .
        .
<code for the data interface
```

```
(set parameters, gather data)>
            .
            .
$NPFCHK TYP=CALL,...
<call of the corresponding COSMOS routine>

            .
<if required, code to restore the module environment>

            .
$NPFCHK TYP=END
<end of measuring point>
```

User Description

Compatibility:

Source compatibility is guaranteed for at least two consecutive BS2000 versions.

Usage Restrictions:

FL = TPR , SIH
FL = TPR : all parameters can be used
FL = SIH : the parameter STATE=II in
$NPFCHK TYP=CALL is forbidden!
Domain=COSMOS

Performance:

If the parameter EXVT=YES is given in the TYP=CALL macro one statement less is used for each measuring point as with EXVT=NO. Therefore this parameter should always be used if EXVT coverage exists at the measuring point in question.

If the parameter CHECK=NSTD is not used at the TYP=BEGIN two statements more will be generated for each measuring point. Events located in dynamically loaded components or read only code cannot use this parameter!

The use of this parameter is only allowed in co-ordination with the COSMOS-Team. For some events that already exist COSMOS assumes CHECK=NSTD.

If the parameter STATE=IN is given at TYP=CALL (uninterruptability will be set and kept inside the measuring point) three statements more will be used as for STATE=NN. STATE=II (measuring point interruptible) increases the number of statements by about 30.

Macro Syntax:

The MF= parameter is not supported

Operand Description:

1 Expansion of a short macro description

NAME $NPFCHK INFO=ALL

for the calling sequence and a description of all three macro types or

NAME $NPFCHK INFO=YES,TYP= BEGIN | CALL | END

for the description of one type.

2 $NPFCHK TYP=BEGIN

```
NAME $NPFCHK TYP=BEGIN, EV#=NNN ,EVNAME=CCCC
,WREG1=R1 ,WREG2=R2
,SBEV#=NNN ,SBEVNAM=CCCC
,HOOKID=A ,CHECK= STD | NSTD
```

NAME = free label (assembler label)

TYP=BEGIN macro type

EV# = event number; a number divisible by 4 without rest greater equal 12 This number is known for already existing events For new events it must be defined in co-ordination with the developer responsible for COSMOS

EVNAME = event name; 4 characters (not yet evaluated)

WREG1 = a general register, pre-set to R15 WREG1 is a work register which can be used by $NPFCHK. If it is not given COSMOS assumes GR 15 can be used.

WREG2 = a general register The parameter WREG2 is only supported for compatibility reasons but will no longer be evaluated.

HOOKID = 1 character (alphanumeric) Pre-set to " ".

The definition of the hookid must be co-ordinated with the developer responsible for COSMOS.

SBEV# = subevent number; number between 1 and 8 Its use has to be co-ordinated also with the developer responsible for COSMOS.

SBEVNAM = subevent name; 4 characters (not yet evaluated)

CHECK = STD a bit in a system table will be tested to decide whether the monitor should be called (pre-set for events with event numbers greater than 144)

= NSTD the old method (transformation of a branch to a nop) will be used to call the monitor (pre-set for events with event numbers 16,20,24,28,44,48,84,100).

The parameters TYP and EV# are mandatory.

3 $NPFCHK TYP=CALL
NAME $NPFCHK TYP=CALL ,EXREG=R1 ,RETREG=R2
,SEQCHK= YES | NO
,EXVT= YES | NO
,STATE= NN | IN | II
,SVAREA= (13) | (R3) | ADDR

TYP=CALL macro type

EXREG = a general register; calling register for the COSMOS call, pre-set to R15. Will not be evaluated if STATE=II is set (R15 will always be used then).

RETREG = a general register; return register from COSMOS call, pre-set to R14. Will not be evaluated if STATE=II is set (R14 will always be used then).

SEQCHK = YES the existence of a previous call of the macro with TYP=BEGIN will be checked (pre-set).

= NO the check will not be done

EXVT = YES the EXVT is addressable in this part of the code

= NO the EXVT will be addressed by a V-constant (pre-set)

STATE = NN the routine where the macro is called is not interruptible

= IN the uninterruptibility will be obtained inside the measuring point

= II the code is interruptible (must not be given if FL = SIH !!!)

SVAREA = (13) general register 13 contains the address of a 18 FW save area (only needed with STATE=II) , pre-set

= (R3) the register contains the address of a 18 FW save area it will be loaded into R13 only needed with STATE=II)

ADDR a 18 FW save area is at this address the address will be loaded into R13 (only needed with STATE=II)

The parameter TYP is mandatory.

Additionally the same parameters as with TYP=BEGIN can be given (i.e. EVNAME, EV#, HOOKID, SBEV#, SBEVNAM).
This possibility is used if the TYP=CALL invocation of the macro is located outside of the BEGIN-END sequence, maybe even in another module (see also the parameter SEQCHK).

For reasons of clearness of the structure and error proneness the use of this possibility should severely be restricted and preference should be given to realizing the sequence BEGIN, CALL, END in one module.

Programming Note:
The call of the macro $NPFCHK TYP=CALL, STATE=II, ... destroys the registers R2, R3, R14 and R15 (if SVAREA is not equal (13) R13 also).

4 $NPFCHK TYP=END
NAME $NPFCHK TYP=END ,SEQCHK= YES | NO

TYP=END macro type

SEQCHK = YES the existence of a previous call of the macro with TYP=CALL will be checked pre-set

= NO the check will not be done.

The parameter type is mandatory.

10.3 $NPCHKI(SPL,) - Decoupling Interface to Insert COSMOS Hooks*

User Specification:
$NPCHKI provides interfaces for system programs
1 to declare (sub)events
2 to define the basic structure of COSMOS parameters in a MODEL or DSECT
3 to check whether the (sub)event is opened by the monitor
4 to call the monitor

Compatibility:
object level since V10

Usage Restriction:
FL = TPR | SIH
SL = RESTRICTED-COSMOS

COSMOS measuring points (so called hooks) are introduced in system modules to connect them to the monitor COSMOS. The identification of a hook is given by event name and subevent name. These values are defined in co-ordination with the COSMOS developer.

One of the parameters for the monitor call points to a user data record that is output to COSMOS file if the monitor is running and the declared (sub)event is enabled (opened) by the monitor. The structure of the user data record must also be defined in co-ordination with the COSMOS developer because the evaluators CAP and NEGET (interface modules for COSMOS files) have to know it.

GC Types:
Conform to convention *Schnittstelleneinschalung* (SDPF 500.22.13).

Language Particularities concerning MF:
SPL: Supported-MF-Values = (D,E,I)
ASSEMBLER: Supported-MF-Values = (C,D,E,L)

GC Syntax:

Name	Operation	Operands
SPL: not appl ASSEMBLER: label	SPL: %USE $NPCHKI ASSEMBLER: $NPCHK	MF=E,PARAM=<param> ,FCT=<fct> ,EVENT=<event> ,SUBEV=<subev> MF=E,PARAM=<param> ,FCT=<fct> ,EVENT=<event> ,SUBEV=<subev> ,INT=YES| NO

Language Particularities concerning Syntax:

Language	Language Specific Operands
SPL:	MF=D [,PREFIX=<prefix>],[ENTRY=YES | NO] MF=I [,PREFIX=<prefix>] ,EVENT=<event> [,SUBEV=<subev>] [,DATAPT=<datapt>]
ASSEMBLER:	MF=C[,PREFIX=<prefix>] MF=D[,PREFIX=<prefix>] MF=L[,PREFIX=<prefix>] ,EVENT=<event> ,SUBEV=<subev> [,DATAPT=<datapt>]

Input Operand Description:

FCT : TEST | CALL
TEST - check whether the monitor is running and the declared (sub)event is enabled.
result : condition code of TM-instruction
CALL - call monitor.
mandatory

DATAPT : pointer to a user data record written to COSMOS file in FCT=CALL if the (sub)event is enabled. mandatory for FCT=CALL
The first two bytes of the data must contain the length of the data (inclusive length field). The maximum length is 240 bytes. The next two bytes specify the version of the data layout. All user data with same version number must have unique form.

EVENT : event name = one of the values declared in the set OSEVENT (SPL).
Assembler : name without pNPC !
mandatory

SUBEV : subevent name = one of the values declared in the set COSSUBEV (SPL).
Assembler : name without pNPC !
Default value : NPCSEV1 (SPL)
SEV1 (Assembler)

Language Particularities concerning Input Operands:

SPL

PREFIX
prefix of all non qualified names for models and sets
default value : $NPCHK

PARAM
name of parameter list

mandatory

ENTRY
YES - entry declarations are generated
NO - no entry declaration is generated
default value : YES

Definition of parameter list and names :

```
%USE $NPCHKI MF=D,PREFIX=prefix,ENTRY=YES;

DCL prefix_COSEVENT SET
      ( nam1 = e1, ....
        namk = ek );

DCL prefix_COSSUBEV SET
```

```
          ( NPCSEV1 = s1, ....
            NPCSEV8 = s8 );
      DCL 1 prefix_PL_MDL   MODEL BOUNDARY(FW)     ,
            2 HEADER         MODE (ESMFHDR )      ,
          /* EVENT_DEF defines the hook                          */
            2 EVENT_DEF                        ,
              3 EVENT        CLASS prefix_COSEVENT (15) ,
              3 SUBEVENT     INTEGER             (8)  ,
              3 RESERVED     INTEGER             (8)  ,
          /* DATAPT points to the user data record (only for FCT=CALL) */
            2 DATAPT         POINTER             ,
          /* reserved field                                      */
            2 RESERVED_INFO   BIN FIXED           (31) ;

        DCL NPFCTAB ENTRY EXTERNAL;

        DCL NPCHIC ENTRY EXTERNAL ( MODE ( prefix_PL_MDL ) );
```

Assembler

label
name of first statement

PREFIX
first letter of all names
default value : N

PARAM
address of parameter list
<param> = <addr> | (<reg>)
<param> = <addr> : <addr> specifies the name of the parameter list
<param> = (<reg>): register <reg> contains address of parameter list

INT
interruptibility during monitor call
YES : coding is interruptible
NO : coding is non-int.
default value : YES

Definition of parameter list and names :

```
label    $NPCHK MF=D,PREFIX=p

  pNPCnam1 EQU   e1
         ...
  pNPCnamk EQU   ek
  pNPCSEV1 EQU   s1
         ...
  pNPCSEV8 EQU   s8


  label    DSECT
  pNPCHDR  FHDR  MF=(D,p)
  pNPCEV   DS    0A              event decl.
  pNPCEVE  DS    Y               event
  pNPCEVS  DS    AL1               subevent
  pNPCEVR  DS    AL1               reserved
  pNPCPT   DS    A               data pointer if FCT=CALL
  pNPCRI   DS    F               reserved
  pNPC#    EQU   *-pNPCHDR
```

Output Description:
Return information after calling $NPCHK,FCT=TEST :
condition code (CC) !!!

CC = ZERO : the hook is opened, i.e. the monitor is running and monitor calls by $NPCHK FCT=CALL are possible
CC not ZERO : the hook is closed

Register Usage at Branch-Level within GC:
INT=YES : ISL-Linkage
INT=NO : call by ##BALR R14,R15
R1 : addr. of parameters

Entry-Names:
NPCHIC
NPCHNC
NPFCTAB

Return Codes:

SC2	name	SC1	name	MAIN	name	meaning
0	-	00	NNPCS1OK	0000	NPCHMOK	normal processing
0	-	40	NNPCS1EE	0005	NPCHMUE	unknown event
0	-	40	NNPCS1EE	0006	NPCHMUS	unknown subevent
0	-	40	NNPCS1EE	0007	NPCHMIP	invalid pointer to user data record
0	-	40	NNPCS1EE	0008	NPCHMIL	invalid length of user data record

Programming Notes:

1 $CNTR is called in routine NPCHIC (FCT=CALL,INT=YES). $CNTR locks the own CPU.

2 For performance reasons the call of the monitor (and data gathering) should be done only if the result of $NPCHK(I) FCT=TEST shows that the declared (sub)event is enabled. If the monitor is called and the (sub)event is disabled (closed) then no data are gathered by the monitor and the call has no effect.

3 Layout of a user data record :
- 3.1 2 byte length field (length inclusive length field and version field, maximum : 240 bytes)
- 3.2 2 byte version of user data record layout (integer)
- 3.3 user data record

4 The parameter INT=xxx is only available in the assembler macro.

5 For INT=NO the address of the EXVT is loaded into R11 in the called routine (also during the call of the monitor event handlers). R11 is restored on return to the hook. R15 is modified after return.

6 The expansion of FCT=TEST generates a TM-instruction !!! The condition code shows whether the (sub)event is open (see example). This is necessary for performance reasons. The same event name and subevent name must be given in MF=E,FCT=TEST and MF=L.

Examples:

SPL

```
    /* declaration of user data                          */
  DCL 1 DATA_MODEL   MODEL BOUNDARY(FW)  ,
        2 DATA_LNG      BIN FIXED   (15) ,
        2 DATA_VERSION  BIN FIXED   (15) ,
        2 DATA          ......          ;
  DCL MY_DATA  MODE ( DATA_MODEL ) ;

    /* declaration of model and event names              */
  %USE $NPCHKI MF=D;

    /* declaration and initialization of parameter list  */
  DCL HOOK_PAR  MODE ( $NPCHK_PL_MDL ) INIT
     ( %USE $NPCHKI MF=I, EVENT=nami,
                DATAPT=ADDR(MY_DATA) ; );

    /* ..... program code .....                          */

    /* check whether hook is opened                      */
  IF
  %USE $NPCHKI MF=E,FCT=TEST,EVENT=nami;
  THEN DO;  /* hook is open, call monitor                */
    MY_DATA.DATA_LNG     = SPACE ( MY_DATA ) ;
    MY_DATA.DATA_VERSION = 1 ;
    MY_DATA.DATA         = .....
    %USE $NPCHKI MF=E,PARAM=HOOK_PAR,FCT=CALL;;
  END;   /* call monitor                               */

    /* ..... program code .....                          */
```

Assembler

```
  *       declarations
  HOOKPAR  $NPCHK MF=D
          ...
  PROCI    @ENTR TYP=I,ENV=SPLSPEC,LOCAL=STACK
          ...
          LA    R2,PARLIST
          @DATA BASE=R2,DSECT=HOOKPAR
          MVC   HOOKPAR(NNPC#),INITPL    init params
          @IF   NZ
          $NPCHK MF=E,FCT=TEST,EVENT=nami
          @THEN ,                 hook is open, call monitor
          ...                    gather data for MYDATA
          $NPCHK MF=E,PARAM=(R2),FCT=CALL
          @BEND ,                 call monitor
          ...
  *       data
  INITPL   $NPCHK MF=L,EVENT=nami,DATAPT=MYDATA
  *
  MYDATA   DS    0F
          DC    Y(MYDATAL)        length
          DC    Y(1)              version data layout
          DC    ....              data
  MYDATAL  EQU   *-MYDATA
  *
  STACK    @PAR  D=YES
          ...
          DS    0F
```

```
PARLIST  DS    XL(NNPC#)        dyn. param area
STACK    @PAR  LEND=YES
         ...
```

10.4 $NPCHK(ASS,) - Decoupling Interface to Insert COSMOS Hooks*

For the interface description see in description of $NPCHKI(SPL,*).

10.5 $COSMOSC(CPP,) - Insert COSMOS Hook*

User Specification:

Purpose:

$COSMOS provides interfaces for system programs

1. to declare events
2. to define the basic structure of COSMOS parameters in a MODEL or DSECT
3. to check whether the event is opened by the monitor
4. to call the monitor

Interface type:

CALL

Remarks:

COSMOS measuring points (so called hooks) are introduced in system modules to connect them to the monitor COSMOS. The identification of a hook is given by event name. The event name is defined in co-ordination with the COSMOS developer.

One of the parameters for the monitor call points to a user data record that is output to COSMOS file if the monitor is running and the declared event is enabled (opened) by the monitor.

The structure of the user data record must also be defined in co-ordination with the COSMOS developer because the evaluators CAP and NEGET (interface modules for COSMOS files) have to know it.

Compatibility:

object level since COSMOS V12.0

source level since COSMOS V12.0

Usage Restriction:

Functional Level:
FL=TPR | SIH

Supporting Domain:
DOMAIN=COSMOS

GC Types:

Conform to convention *Schnittstelleneinschalung* (SDPF 500.22.13).

Language Particularities concerning MF:
CPP: Supported-MF-Values = (D,M,I,E)
SPL: Supported-MF-Values = (D,M,I,E)
ASSEMBLER: Supported-MF-Values = (C,D,L,M,E)

GC Syntax:

Name	Operation	Operands
CPP: SPL: not appl. ASSEMBLER:: label	CPP: %USE $COSMOSC SPL: %USE $COSMOSI ASSEMBLER: $COSMOS	MF=E,EVENT=<event> ,PARAM=<param> ,FCT=<fct> ,FL=*TPR|*SIH MF=E,EVENT=<event> ,PARAM=<param> ,FCT=<fct> ,FL=*TPR|*SIH MF=E,EVENT=<event> ,PARAM=<param>

		,FCT=<fct> ,FL=*TPR|*SIH

Language Particularities concerning Syntax:

CPP:	**MF=D** ,FL=*TPR|*SIH ,PREFIX=<prefix> **MF=I** ,PREFIX=<prefix> ,EVENT=<event> ,DATALENGTH=<datalength> ,DATAVERS=<datavers> ,DATAPT=<dataptr> ,TCBPTR=<tcbptr> **MF=M** ,PREFIX=<prefix> ,EVENT=<event> ,DATALENGTH=<datalength> ,DATAVERS=<datavers> ,DATAPT=<dataptr> ,TCBPTR=<tcbptr>
SPL:	**MF=D** ,FL=*TPR|*SIH ,PREFIX=<prefix> ,ENTRY=YES|NO **MF=I** ,PREFIX=<prefix> ,EVENT=<event> ,DATALENGTH=<datalength> ,DATAPT=<dataptr> ,DATAVERS=<datavers> ,TCBPT=<tcbptr> **MF=M** ,PREFIX=<prefix> ,EVENT=<event> ,DATALENGTH=<datalength> ,DATAPT=<dataptr> ,DATAVERS=<datavers> ,TCBPT=<tcbptr>
ASSEMBLER:	**MF=C** ,PREFIX=<prefix> **MF=D** ,FL=*TPR|*SIH ,PREFIX=<prefix> **MF=L** ,PREFIX=<prefix> ,EVENT=<event> ,DATALENGTH=<datalength> ,DATAVERS=<datavers> ,DATAPT=<dataptr> ,TCBPT=<tcbptr> **MF=M** ,PREFIX=<prefix> ,EVENT=<event> ,DATALENGTH=<datalength> ,DATAVERS=<datavers> ,DATAPT=<dataptr> ,TCBPT=<tcbptr>

Input Operand Description:

FCT : *TEST *CALL
*TEST - check whether the monitor is running and the declared event is enabled.
result : condition code of TM-instruction

*CALL - call monitor.
mandatory

FL : *TPR *SIH
only for FCT=*CALL
*TPR - call the entry for TPR
*SIH - call the entry for SIH
default value : *TPR

EVENT : event name = one of the values declared in the set prefix_COSEVENT (SPL).
Assembler : name without pNPC !
mandatory

DATALENGTH : length of the user data. The maximum length of the data is 240 bytes.
mandatory

DATAPT : pointer to a user data record written to COSMOS file in FCT=*CALL if the event is enabled.
mandatory for FCT=*CALL

DATAVERS : version number of the user data. All user data with same version number must have unique form.
mandatory

TCBPT : pointer to the TCB of the task who is generating the event.
If the pointer is not specified, the TIC is used.

Language Particularities concerning Input Operands:

CPP

PREFIX prefix of all non qualified names for data declarations and definitions
default value : sCOSMOS

PARAM name of parameter list
mandatory

ENTRY YES - entry declarations are generated
NO - no entry declaration is generated
default value : YES

Definition of parameter list and names :

```
$USE $COSMOSC MF=D,PREFIX=prefix,ENTRY=YES;

#define prefixnam1   e1
#define prefixnam2   e2
  .
  .
#define prefixnamk   ek
```

```
struct prefix_PL_MDL
      struct ESMFHDR HDR;
      unsigned short EVENT_NUMBER;
      short DATA_LNG;
      void* DATAPTR;
      unsigned short DATA_VERSION;
      short RESERVED1;
      void* TCBPTR;
     ;

extern ISL void NPCHOOK (struct prefix_PL_MDL&);

extern unsigned char (*NPFCTAB) ();
```

SPL

PREFIX prefix of all non qualified names for models and sets
default value : $COSMOS

PARAM name of parameter list
mandatory

ENTRY YES - entry declarations are generated
NO - no entry declaration is generated
default value : YES

Definition of parameter list and names :

```
%USE $COSMOSI MF=D,PREFIX=prefix,ENTRY=YES;

DCL   prefix_COSEVENT SET
        ( nam1 = e1, ....
          namk = ek );
DCL 1 prefix_PL_MDL   MODEL BOUNDARY(FW)     ,
      2 HEADER         MODE (ESMFHDR )      ,
      2 EVENT_NUMBER    CLASS prefix_COSEVENT (15) ,
      2 DATA_LNG        BIN FIXED            (15) ,
    /* DATAPTR points to the user data record (only for FCT=CALL) */
      2 DATAPTR         POINTER              ,
      2 DATA_VERSION    INTEGER              (15) ,
    /* reserved field                                */
      2 RESERVED1       BIN FIXED            (15) ,
      2 TCBPTR          POINTER              ;

DCL NPFCTAB ENTRY EXTERNAL;

DCL NPCHOOK ENTRY EXTERNAL ( MODE ( prefix_PL_MDL ) );
```

Assembler

label name of first statement

PREFIX first letter of all names
default value : N

PARAM address of parameter list
<param> = <addr> (<reg>)
<param> = <addr> : <addr> specifies the name parameter list
<param> = (<reg>): register <reg> contains address of parameter list

Definition of parameter list and names :

```
label    $COSMOS MF=D,PREFIX=p

pNPCnam1 EQU   e1
         ...
pNPCnamk EQU   ek


label    DSECT
pPCHHDR  FHDR  MF=(D,p)
pPCHEV#  DS    Y              event number
pPCHLNG  DS    Y              data length
pPCHPTR  DS    A              data pointer if FCT=CALL
pPCHVERS DS    Y              version number of user data
pPCHRES1 DS    Y              reserved
pPCHTCBP DS    A              pointer to TCB
pPCH#    EQU   *-pPCHHDR
```

Output Description:

Return information after calling $COSMOS,FCT=*TEST :

not ZERO : the hook is opened, i.e. the monitor is running and monitor calls by $COSMOS FCT=*CALL are possible
ZERO : the hook is closed

Procedure Linkage:

FL=*TPR : DSL-Linkage
FL=*SIH : call by ##BALR R14,R15
R1 : addr. of parameter-list

Entry-Name(s) or SVC-Number(s):

NPCHOOK
NPCHSIH

Return Codes:

SC2	SC1	MAIN	Name	Meaning
0	0	0	NPCHMOK	normal processing
0	40	5	NPCHMUE	unknown event
0	40	7	NPCHMIP	invalid pointer to user data record
0	40	8	NPCHMIL	invalid length of user data record

Programming Notes:

1. For performance reasons the call of the monitor (and data gathering) should be done only if the result of $COSMOS(I) FCT=*TEST shows that the declared event is enabled. If the monitor is called and the event is disabled (closed) then no data are gathered by the monitor and the call has no effect.
2. If the layout of the data record is changed the data version must be incremented.
3. The expansion of FCT=*TEST generates inline code !!!
   The condition code shows whether the event is open (see example). This is necessary for performance reasons.
   The same event name must be given in MF=E, FCT=*TEST and MF=L.

Example:

CPP

```
  /* definition of user data                          */
struct
  ...    ;
   params;

$USE FHDRC MF=D;
$USE $COSMOSC MF=D;

struct sCOSMOS_PL_MDL   HOOK_PAR;

$USE $COSMOS MF=I, PARAM=HOOK_PAR, DATAVERS=1, EVENT=nami,
          DATAPT=&(params),DATALENGTH=sizeof(params);

  /* ..... program code ........                      */


  /* check whether hook is opened                     */

if $USE $COSMOSC MF=E, FCT=*TEST, EVENT=nami;
   /* hook is open, call monitor */
    ....
    $USE $COSMOSC MF=E, FCT=*CALL, PARAM=HOOK_PAR;
   ;
```

SPL

```
  /* declaration of user data                         */
DCL 1 DATA_MODEL   MODEL BOUNDARY(FW)  ,
    2 DATA         ......          ;
DCL MY_DATA  MODE ( DATA_MODEL ) ;

  /* declaration of model and event names             */
%USE $FHDRI  MF=D;
%USE $COSMOSI MF=D;

  /* declaration and initialization of parameter list */
DCL HOOK_PAR  MODE ( $COSMOS_PL_MDL ) INIT
   ( %USE $COSMOSI MF=I, EVENT=nami,
            DATALENGTH=SPACE(MY_DATA), DATAPT=ADDR(MY_DATA),
            DATAVERS=1 ;);

  /* ..... program code .....                         */

  /* check whether hook is opened                     */
IF
%USE $COSMOSI MF=E,FCT=*TEST,EVENT=nami;
THEN DO;  /* hook is open, call monitor               */
  MY_DATA.DATA        = .....
  %USE $COSMOSI MF=E,PARAM=HOOK_PAR,FCT=*CALL;;
END;   /* call monitor                                */

  /* ..... program code .....                         */
```

Assembler

```
*       declarations
HOOKPAR  $COSMOS MF=D
        ...
PROCI    @ENTR TYP=I,ENV=SPLSPEC,LOCAL=STACK
        ...
```

```
        LA    R2,PARLIST
        @DATA BASE=R2,DSECT=HOOKPAR
        MVC   HOOKPAR(NPCH#),INITPL    init params
        @IF   NZ
        $COSMOS MF=E,FCT=*TEST,EVENT=nami
        @THEN ,                hook is open, call monitor
        ...                    gather data for MYDATA
        $COSMOS MF=E,PARAM=(R2),FCT=*CALL
        @BEND ,                call monitor
        ...
*       data
INITPL  $COSMOS MF=L,EVENT=nami,DATAPT=MYDATA,DATAVERS=1   -
          DATALENGTH=MYDATAL
*
MYDATA  DS    0F
        DC    ....             data
MYDATAL EQU   *-MYDATA
*
STACK   @PAR  D=YES
        ...
        DS    0F
PARLIST DS    XL(NPCH#)        dyn. param area
STACK   @PAR  LEND=YES
        ...
```

10.6 $COSMOS(ASS,) - Insert COSMOS Hook*

For the interface description see in description of $COSMOSC(CPP,*).

10.7 $COSHPV(ASS,) - Call-Interface between COSMOS and VM2000*

User Specification:

Purpose:
INTERFACE TO WRITE VM2000-EVENTS INTO COSMOS-FILE

Interface type:
CALL

Remarks:
The use of this interface is restricted to VM2000.

Compatibility:
object level COSMOS since V12.0
source level COSMOS since V12.0

Usage Restriction:

Functional Level:
FL=TPR

Supporting Domain:
DOMAIN=COSMOS

Restrictions of Direct Interface Usage in Sources and GCs:
The usage of the interface may only be directly coded in sources of VM2000.

GC Types:
As this interface must only be used by VM2000, it is sufficient to maintain its ASSEMBLER language form.

Language Particularities concerning MF:

CPP: None

SPL: None

ASSEMBLER: Supported-MF-Values = (D, C, L, M, E)

GC Syntax:

$COSHPV
Operands
DATAPT = <var: pointer> DATALENGTH = <integer 1..32767> | var: int:2>
$COSHPV

Control operands MF and PARAM are supported according to convention.

Language Particularities concerning Syntax:

Language	Language specific operand
ASSEMBLER	\PREFIX=N\ \MACID=PCV>

Operand Description:

DATAPT: Functions-Operand
Pointer to buffer with HPV-events
mandatory

DATALENGTH: Functions-Operand
Length of buffer with HPV-events
mandatory

Operand Conditions:

None.

Parameter Area Description:

Offset	Identifier		Data type or value	Meaning
FW	$COSHPV_pl_mdl,	ASS:NPCVMDL	STRUCT:16	PARAMETER LIST LAYOUT
000	hdr		ESMFHDR:8	standard header
008	dataprt	,ASS:NPCVPTR	PTR:4	buffer pointer
00C	data_lng	,AS:NPCVLNG	INT:2	buffer length
00E	reserved1	,ASS:NPCVRES1	INT:1	reserved

Procedure Linkage:

ISL

Entry-Name(s) or SVC-Number(s):

NPCVM2

Return Code:

SRC2	SCR1	MRC	Identifier for MRC		Meaning
00	00	0000	no_error	,ASS:NPCVMOK	normal processing
00	40	0007	ptr_error	,ASS:NPCVMIP	invalid dataptr
00	40	0008	lng_error	,ASS:NPCVMIL	invalid data length

Example**:**

Assembler

```
*       declarations
            PARLD   $COSHPV MF=D
                ...
            PROCI   @ENTR TYP=I,ENV=SPLSPEC,LOCAL=STACK
                ...
                LA    R2,PARLIST
                @DATA BASE=R2,DSECT=PARLD
                MVC   PARLD(NPCV#),INITPL    init params
```

```
        $COSHPV MF=M,DATAPT=BUFFPTR,DATALENGTH=BUFFLEN
        ...                 gather HPV-events
        $COSHPV MF=E,PARAM=(R2)
        ...
*       data
INITPL  $COSHPV MF=L
*
BUFFPTR DS   0F
        DC   ....           data
BUFFLEN EQU  *-BUFFPTR
*
STACK   @PAR D=YES
        ...
        DS   0F
PARLIST DS   XL(NPCV#)      dyn. param area
STACK   @PAR LEND=YES
        ...
```

10.8 $COSBUF(ASS,) - Data-Interface between COSMOS and VM2000*

User Specification:

Purpose:
This GC generates the data layout of the COSMOS-buffer header and the COSMOS-event header.

Interface type:
LAYOUT

Remarks:
The use of these data packets is restricted to VM2000.

Compatibility:
object level COSMOS since V12.0
source level COSMOS since V12.0

Usage Restriction:

Functional Level:
FL=TPR

Supporting Domain:
DOMAIN=COSMOS

Restrictions of Direct Interface Usage in Sources and GCs:
The usage of the interface may only be directly coded in sources of COSMOS and VM2000.

GC Types:
As this interface must only be used by VM2000, it is sufficient to maintain its Assembler language form.

Language Particularities concerning MF:

CPP:	None.
SPL:	None.
ASSEMBLER:	Supported-MF-Values = (C, D)

GC Syntax:
The following COSMOS data descriptions are provided:
- BUFHDR buffer header
- EVHDR event header

Each data structure can be achieved by the specification of the data identifier at the macro call.
Control operands MF=D and MF=C are only accepted.

Language particularities concerning syntax:

Language	Language specific operands
ASSEMBLER	\PREFIX=N\ \MACID=PCE\

10.8.1 EVTHDR - Event header data description

$COSBUF

Operands

DATA = <c-string 1..6>

$COSBUF

Operand Description:

DATA**:** Control-Operand
Specifies the data description
EVTHDR c-string (1,6)

Operand Conditions:

DATA=EVTHDR

Data Area Description:

```
*   EVENT HEADER LAYOUT
NPCEVHDR DSECT ,
NPCEIND  DS    X         indic. for begin of event
NPCESTMP DS    XL4       time stamp (second word of STCK)
NPCERES1 DS    X         reserved for format of time stamp
NPCELN   DS    X         event length (NPCEVLN + length of data area)
NPCEEV#  DS    XL2       event code
NPCETSN  DS    CL4       TSN
NPCEVERS DS    XL2       version of data
NPCELM#  DS    XL2       logical machine number
NPCEVHDR# EQU  *-NPCEIND  total length of header
NPCEVLN  EQU   *-NPCEEV#  length from event code
NPCEDATA DS    0X        start of data area
```

10.8.2 BUFHDR - Buffer header data description

$COSBUF

Operands

DATA = <c-string 1..6>

$COSBUF

Operand Description:

DATA:
Control-Operand
Specifies the data description
BUFHDR c-string (1,6)

Operand Conditions:

DATA=BUFHDR

Data Area Description:

```
    *   BUFFER HEADER LAYOUT
    NPCECFID DS    F                  cosfil file-id
    *
    NPCETS   DS    0XL8               time stamp LAYOUT (STCK-format)
    *                           the time stamp of the first
    *                           event in buffer is expected
    NPCETS1  DS    F                  first part
    NPCETS2  DS    F                  second part
    *
    NPCEME1  DS    F                  missed events lm1
    NPCEME2  DS    F                  missed events lm2
    NPCEME3  DS    F                  missed events lm3
    NPCEME4  DS    F                  missed events lm4
    NPCEME5  DS    F                  missed events lm5
    NPCEME6  DS    F                  missed events lm6
    NPCEME7  DS    F                  missed events lm7
    NPCEME8  DS    F                  missed events lm8
    NPCECSQ  DS    H                  nr. bufferptr. has changed
    NPCESAQ  DS    H                  nr. startaddr. beyond bounds
    NPCESLTQ DS    H                  missed events due to slots
    NPCE#    EQU   *-NPCECFID
```

Programming Notes

In the buffer header only the time stamp had to be completed by VM2000.

10.9 $COSSM2 - Call-Interface between COSMOS and SM2

Interface name :
$COSSM2

Interface name(ASS):
$COSSM2

Interface name(CPP):

Interface name(SPL):
$COSSM2I

Interface title :
Interface $COSSM2

Interface domain :
COSMOS

Compilation scope :
RESTRICTED

Interface type :
CALL

Run context :
TPR

Interface purpose :
Interface between COSMOS and SM2

Interface remarks :
UNSPECIFIED

Language :
ASS,SPL

Prefix :
N

Macid :

PFS

Syntax SPL:

```
                                        $COSSM2I
Operands

ENTRY       =     YES |
                  NO
FUNCT       =     *INIT |
                  *START |
                  *STOP |
                  *DEINIT

                                        $COSSM2I
```

Syntax ASS:

```
                                        $COSSM2I
Operands

FUNCT       =     *INIT |
                  *START |
                  *STOP |
                  *DEINIT

                                        $COSSM2I
```

MF-specials-SPL: D, E, M, I

MF-specials-ASS: C, D, E, L, M,

Operand description :

ENTRY:
- YES
- NO

FUNCT: function of SM2-Call

FUNCT=4	INTITIA COSMOS
FUNCT=5	START COSMOS
FUNCT=6	STOP COSMOS
FUNCT=8	DEINIT COSMOS

*INIT	initiate COSMOS
*START	start COSMOS
*STOP	stop COSMOS
*DEINIT	deinitiate COSMOS

Operand conditions :

Data declarations for SPL:

```
:* main return codes
 DCL %PREFIX"MRET_CODE SET (
    SUCCESS=0                    :* no error detected
   , COSM_ACT=1                   :* COSMOS already active
   , TASK_SEL=2                  :* warning during
                          :* task-selection
   , MAC_N_TERM=3                    :* error during
                          :* macro processing; no term
   , MAC_TERM=4                    :* error during
                          :* macro processing; term
   , DMS_ERR=5                   :* COSMOS-DMS error
   , TLT_ERR=6                  :* TLT error
```

```
   , BTASK_ERR=7                    :* activate btask error
   , UTM_ERR=8                      :* error during START-SS UTM
);

DCL %PREFIX"INIT INTEGER(8) CONSTANT(2);
                                    :* init COSMOS

DCL %PREFIX"START INTEGER(8) CONSTANT(3);
                                    :* start COSMOS

DCL %PREFIX"STOP INTEGER(8) CONSTANT(4);
                                    :* stop COSMOS

:* hook table entry
DCL 1 %PREFIX"HOOK_TABLE_ENTRY MODEL
   , 2 EVT_NUMBER INTEGER(15)       :* event number of entry
   , 2 SUBEVT_MASK INTEGER(8)       :* subevent mask of this event
   , 2 INDICATOR INTEGER(8)         :* indicator of this event
;

:* parameter area description
DCL 1 %PREFIX"PARM MODEL
   , 2 HDR MODE(ESMFHDR)            :* Standard header
   , 2 ERROR_TEXT CHAR(80)          :* error text
   , 2 SECOND_RETCOD INTEGER(31)    :* return codes from macros
                                    :* used by COSMOS
   , 2 TITLE CHAR(80)               :* title of measurement
   , 2 NR_FILES INTEGER(8)          :* number of files
   , 2 IND                          :* indicator for writing
      , 3 UNUSED_1 BIT(3)           :* unused
      , 3 WRAP_AROUND BIT(1)        :* writing wrap around
      , 3 STREAM BIT(1)             :* write files in streaming
                                    :* mode
      , 3 NOCONF BIT(1)             :* write no configuration into
                                    :* file
      , 3 INFO BIT(1)               :* type all p2parms
      , 3 UNUSED_2 BIT(1)           :* unused
   , 2 NR_BUFFERPAGES INTEGER(8)    :* number pages of buffer
   , 2 NR_BUFFER INTEGER(8)         :* number of buffer
   , 2 NR_SLOT INTEGER(8)           :* number of slots
   , 2 NR_TSN INTEGER(8)            :* number of tsn's
   , 2 TSN_WARN_1                   :* bitlist of tsn's for
                                    :* warnings
      , 3 TSNF_N_FD BIT(1)          :* tsn 16 not found
      , 3 TSNE_N_FD BIT(1)          :* tsn 15 not found
      , 3 TSND_N_FD BIT(1)          :* tsn 14 not found
      , 3 TSNC_N_FD BIT(1)          :* tsn 13 not found
      , 3 TSNB_N_FD BIT(1)          :* tsn 12 not found
      , 3 TSNA_N_FD BIT(1)          :* tsn 11 not found
      , 3 TSN9_N_FD BIT(1)          :* tsn 10 not found
      , 3 TSN8_N_FD BIT(1)          :* tsn 9 not found
   , 2 TSN_WARN_2                   :* bitlist of tsn's for
                                    :* warnings
      , 3 TSN7_N_FD BIT(1)          :* tsn 8 not found
      , 3 TSN6_N_FD BIT(1)          :* tsn 7 not found
      , 3 TSN5_N_FD BIT(1)          :* tsn 6 not found
      , 3 TSN4_N_FD BIT(1)          :* tsn 5 not found
      , 3 TSN3_N_FD BIT(1)          :* tsn 4 not found
      , 3 TSN2_N_FD BIT(1)          :* tsn 3 not found
      , 3 TSN1_N_FD BIT(1)          :* tsn 2 not found
      , 3 TSN0_N_FD BIT(1)          :* tsn 1 not found
   , 2 TSN_LST (16) CHAR(4)         :* list of tsn's
   , 2 NR_USERID INTEGER(8)         :* number of user-id's
   , 2 USERID_WARN                  :* bitlist of user-id's for
                                    :* warnings
      , 3 USID8_N_FD BIT(1)         :* user-id 8 not found
      , 3 USID7_N_FD BIT(1)         :* user-id 7 not found
```

```
    , 3 USID6_N_FD BIT(1)          :* user-id 6 not found
    , 3 USID5_N_FD BIT(1)          :* user-id 5 not found
    , 3 USID4_N_FD BIT(1)          :* user-id 4 not found
    , 3 USID3_N_FD BIT(1)          :* user-id 3 not found
    , 3 USID2_N_FD BIT(1)          :* user-id 2 not found
    , 3 USID1_N_FD BIT(1)          :* user-id 1 not found
  , 2 USERID_LST (8) CHAR(8)       :* list of user-id's
  , 2 NR_JOBNAME INTEGER(8)        :* number of job names
  , 2 JOBNAME_WARN                 :* bitlist of job names for
                                   :* warnings
    , 3 JOBN8_N_FD BIT(1)          :* job name 8 not found
    , 3 JOBN7_N_FD BIT(1)          :* job name 7 not found
    , 3 JOBN6_N_FD BIT(1)          :* job name 6 not found
    , 3 JOBN5_N_FD BIT(1)          :* job name 5 not found
    , 3 JOBN4_N_FD BIT(1)          :* job name 4 not found
    , 3 JOBN3_N_FD BIT(1)          :* job name 3 not found
    , 3 JOBN2_N_FD BIT(1)          :* job name 2 not found
    , 3 JOBN1_N_FD BIT(1)          :* job name 1 not found
  , 2 JOBNAME_LST (8) CHAR(8)      :* list of job names
  , 2 NR_CAT INTEGER(8)            :* number of categories
  , 2 CAT_WARN                     :* bitlist of categories for
                                   :* warnings
    , 3 CAT8_N_FD BIT(1)           :* category 8 not found
    , 3 CAT7_N_FD BIT(1)           :* category 7 not found
    , 3 CAT6_N_FD BIT(1)           :* category 6 not found
    , 3 CAT5_N_FD BIT(1)           :* category 5 not found
    , 3 CAT4_N_FD BIT(1)           :* category 4 not found
    , 3 CAT3_N_FD BIT(1)           :* category 3 not found
    , 3 CAT2_N_FD BIT(1)           :* category 2 not found
    , 3 CAT1_N_FD BIT(1)           :* category 1 not found
  , 2 CAT_LST (8) CHAR(7)          :* list of categories
  , 2 TYPE                         :* task type, setting all
                                   :* task types for *ALL
    , 3 UNUSED_2 BIT(4)            :* unused
    , 3 TP BIT(1)                  :* transaction processing task
    , 3 BAT BIT(1)                 :* batch task
    , 3 DIA BIT(1)                 :* dialog task
    , 3 SYS BIT(1)                 :* system task
  , 2 NR_IC INTEGER(8)             :* number of IC's for
                                   :* EIA-Event
  , 2 IC_LST (16) INTEGER(8)       :* list of IC's for EIA-Event
  , 2 NR_SVC INTEGER(8)            :* number of SVC's for
                                   :* EIA-Even
  , 2 SVC_LST (8) INTEGER(8)       :* list of SVC's for EIA-Event
  , 2 NR_MN INTEGER(8)             :* number of dev-mnemonics for
                                   :* IONQ-, SDV-, CHTM-Event
  , 2 MN_LST (8) CHAR(4)           :* list of dev-mnemonics for
                                   :* IONQ-, SDV-, CHTM-Event
  , 2 NR_DABVSN INTEGER(8)         :* number of device-vsn's for
                                   :* DAB-Event
  , 2 DABVSN_LST (8) CHAR(6)       :* list of device-vsn's for
                                   :* DAB-Event
  , 2 NR_MEMCL INTEGER(8)          :* number of memory classes
                                   :* for REQM- and RELM-Event
  , 2 MEMCL_LST (4) INTEGER(8)     :* list of memory classes for
                                   :* REQM- and RELM-Event
  , 2 NR_SLOTMEM INTEGER(8)        :* number of memory classes
                                   :* for SLOT-Event
  , 2 SLOTMEM_LST (4) INTEGER(8)   :* list of memory classes for
                                   :* SLOT-Event
  , 2 NR_PEND INTEGER(8)           :* number of pend codes for
                                   :* PEND-Event
  , 2 PEND_LST (16) INTEGER(8)     :* list of pend codes for
                                   :* PEND-Event
  , 2 NR_LOCK INTEGER(8)           :* number of lock-id's for
                                   :* LOCK-Event
  , 2 LOCK_LST (4) CHAR(2)         :* list of lock-id's for
```

```
                              :* LOCK-Event
   , 2 NR_TLT INTEGER(8)          :* number of tlt-descr for
                              :* TLT-Event
   , 2 TLT_LST (8) CHAR(3)        :* list of tlt-descr for
                              :* TLT-Event
   , 2 TSKI_SWITCH                :* TSKI switch for TSKI-Event
      , 3 UNUSED_3 BIT(7)          :* unused
      , 3 TSKI_ON BIT(1)           :* TSKI switch on
   , 2 NR_TSVC INTEGER(8)         :* number of SVC's for
                              :* TSVC-Event
   , 2 TSVC_LST (8) INTEGER(8)    :* list of SVC's for
                              :* TSVC-Event
   , 2 EVENT_IND                  :* indicator for special
                              :* events
      , 3 UNUSED_4 BIT(1)          :* unused
      , 3 UTM BIT(1)               :* UTM-Event
      , 3 DCAM BIT(1)              :* DCAM-Event
      , 3 BCAM BIT(1)              :* BCAM-Event
      , 3 LTS BIT(1)               :* LTS-Event
      , 3 BCPT BIT(1)              :* BCPT-Event
      , 3 TINF BIT(1)              :* TINF-Event
      , 3 XEIA BIT(1)              :* XEIA-Event
   , 2 HOOK_TABLE (100) MODE(%PREFIX"HOOK_TABLE_ENTRY)
                              :* table with hooks to be
                              :* opened
;
```

Data declarations for CPP:

Data declarations for ASS:

```
NPFSINIT EQU   4                  init COSMOS
*
NPFSSTART EQU   5                  start COSMOS
*
NPFSSTOP EQU   6                   stop COSMOS
*
*   parameter area description
NPFSPARM DS    0F
NPFSHDR  FHDR  MF=(C,NPFS),EQUATES=NO    Standard header
*   main return codes
NPFSSUCCESS EQU   0                no error detected
NPFSCOSM_ACT EQU   1               COSMOS already active
NPFSTASK_SEL EQU   2               warning during task-selection
NPFSMAC_N_TERM EQU   3             error during macro processing;
*                              no term
NPFSMAC_TERM EQU   4               error during macro processing;
*                              term
NPFSDMS_ERR EQU   5                COSMOS-DMS error
NPFSTLT_ERR EQU   6                TLT error
NPFSBTASK_ERR EQU   7              activate btask error
*
NPFSERROR_TEXT DS    CL80          error text
NPFSSECOND_RETCOD DS    F          return codes from macros used
*                              by COSMOS
NPFSTITLE DS    CL80               title of measurement
NPFSNR_FILES DS    X               number of files
NPFSIND  DS    AL1                indicator for writing
NPFSUNUSED_1 EQU   X'E0'           unused
NPFSWRAP_AROUND EQU   X'10'        writing wrap around
NPFSSTREAM EQU   X'08'             write files in streaming mode
NPFSNOCONF EQU   X'04'             write no configuration into
*                              file
NPFSINFO EQU   X'02'               type all p2parms
NPFSNR_BUFFERPAGES DS    X         number pages of buffer
NPFSNR_BUFFER DS    X              number of buffer
NPFSNR_SLOT DS    X                number of slots
```

```
NPFSNR_TSN DS    X                number of tsn's
NPFSTSN_WARN DS    AL1                bitlist of tsn's for warnings
NPFSTSN8_N_FD EQU   X'80'              tsn 8 not found
NPFSTSN7_N_FD EQU   X'40'              tsn 7 not found
NPFSTSN6_N_FD EQU   X'20'              tsn 6 not found
NPFSTSN5_N_FD EQU   X'10'              tsn 5 not found
NPFSTSN4_N_FD EQU   X'08'              tsn 4 not found
NPFSTSN3_N_FD EQU   X'04'              tsn 3 not found
NPFSTSN2_N_FD EQU   X'02'              tsn 2 not found
NPFSTSN1_N_FD EQU   X'01'              tsn 1 not found
NPFSTSN_LST DS    8CL4               list of tsn's
NPFSTSN_LST# EQU   8
NPFSNR_USERID DS    X                number of user-id's
NPFSUSERID_WARN DS    AL1                bitlist of user-id's for
*                               warnings
NPFSUSID8_N_FD EQU   X'80'             user-id 8 not found
NPFSUSID7_N_FD EQU   X'40'             user-id 7 not found
NPFSUSID6_N_FD EQU   X'20'             user-id 6 not found
NPFSUSID5_N_FD EQU   X'10'             user-id 5 not found
NPFSUSID4_N_FD EQU   X'08'             user-id 4 not found
NPFSUSID3_N_FD EQU   X'04'             user-id 3 not found
NPFSUSID2_N_FD EQU   X'02'             user-id 2 not found
NPFSUSID1_N_FD EQU   X'01'             user-id 1 not found
NPFSUSERID_LST DS    8CL8              list of user-id's
NPFSUSERID_LST# EQU   8
NPFSNR_JOBNAME DS    X                number of job names
NPFSJOBNAME_WARN DS    AL1                bitlist of job names for
*                               warnings
NPFSJOBN8_N_FD EQU   X'80'             job name 8 not found
NPFSJOBN7_N_FD EQU   X'40'             job name 7 not found
NPFSJOBN6_N_FD EQU   X'20'             job name 6 not found
NPFSJOBN5_N_FD EQU   X'10'             job name 5 not found
NPFSJOBN4_N_FD EQU   X'08'             job name 4 not found
NPFSJOBN3_N_FD EQU   X'04'             job name 3 not found
NPFSJOBN2_N_FD EQU   X'02'             job name 2 not found
NPFSJOBN1_N_FD EQU   X'01'             job name 1 not found
NPFSJOBNAME_LST DS    8CL8              list of job names
NPFSJOBNAME_LST# EQU   8
NPFSNR_CAT DS    X                number of categories
NPFSCAT_WARN DS    AL1                bitlist of categories for
*                               warnings
NPFSCAT8_N_FD EQU   X'80'              category 8 not found
NPFSCAT7_N_FD EQU   X'40'              category 7 not found
NPFSCAT6_N_FD EQU   X'20'              category 6 not found
NPFSCAT5_N_FD EQU   X'10'              category 5 not found
NPFSCAT4_N_FD EQU   X'08'              category 4 not found
NPFSCAT3_N_FD EQU   X'04'              category 3 not found
NPFSCAT2_N_FD EQU   X'02'              category 2 not found
NPFSCAT1_N_FD EQU   X'01'              category 1 not found
NPFSCAT_LST DS    8CL7               list of categories
NPFSCAT_LST# EQU   8
NPFSTYPE DS    AL1                task type, setting all
*                               task types for *ALL
NPFSUNUSED_2 EQU   X'F0'               unused
NPFSTP   EQU   X'08'               transaction processing task
NPFSBAT  EQU   X'04'               batch task
NPFSDIA  EQU   X'02'               dialog task
NPFSSYS  EQU   X'01'               system task
NPFSNR_IC DS    X                number of IC's for EIA-Event
NPFSIC_LST DS    16X               list of IC's for EIA-Event
NPFSNR_SVC DS    X                number of SVC's for EIA-Even
NPFSSVC_LST DS    8X               list of SVC's for EIA-Event
NPFSNR_MN DS    X                number of dev-mnemonics for
*                               IONQ-, SDV-, CHTM-Event
NPFSMN_LST DS    8CL4               list of dev-mnemonics for
*                               IONQ-, SDV-, CHTM-Event
NPFSMN_LST# EQU   8
```

```
NPFSNR_DABVSN DS    X                number of device-vsn's for
*                                    DAB-Event
NPFSDABVSN_LST DS    8CL6             list of device-vsn's for
*                                    DAB-Event
NPFSDABVSN_LST# EQU   8
NPFSNR_MEMCL DS    X                 number of memory classes for
*                                    REQM- and RELM-Event
NPFSMEMCL_LST DS    4X               list of memory classes for
*                                    REQM- and RELM-Event
NPFSNR_SLOTMEM DS    X               number of memory classes for
*                                    SLOT-Event
NPFSSLOTMEM_LST DS    4X             list of memory classes for
*                                    SLOT-Event
NPFSNR_PEND DS    X                  number of pend codes for
*                                    PEND-Event
NPFSPEND_LST DS    16X               list of pend codes for
*                                    PEND-Event
NPFSNR_LOCK DS    X                  number of lock-id's for
*                                    LOCK-Event
NPFSLOCK_LST DS    4CL2              list of lock-id's for
*                                    LOCK-Event
NPFSLOCK_LST# EQU   4
NPFSNR_TLT DS    X                   number of tlt-descr for
*                                    TLT-Event
NPFSTLT_LST DS    8CL3               list of tlt-descr for
*                                    TLT-Event
NPFSTLT_LST# EQU   8
NPFSTSKI_SWITCH DS    AL1            TSKI switch for TSKI-Event
NPFSUNUSED_3 EQU   X'FE'             unused
NPFSTSKI_ON EQU   X'01'              TSKI switch on
NPFSNR_TSVC DS    X                  number of SVC's for TSVC-Event
NPFSTSVC_LST DS    8X                list of SVC's for TSVC-Event
NPFSEVENT_IND DS    AL1              indicator for special events
NPFSUNUSED_4 EQU   X'80'             unused
NPFSUTM  EQU   X'40'                 UTM-Event
NPFSDCAM EQU   X'20'                 DCAM-Event
NPFSBCAM EQU   X'10'                 BCAM-Event
NPFSLTS  EQU   X'08'                 LTS-Event
NPFSBCPT EQU   X'04'                 BCPT-Event
NPFSTINF EQU   X'02'                 TINF-Event
NPFSXEIA EQU   X'01'                 XEIA-Event
NPFSHOOK_TABLE DS    0XL4            table with hooks to be opened
NPFSEVT_NUMBER DS    H               event number of entry
NPFSSUBEVT_MASK DS    X              subevent mask of this event
NPFSINDICATOR DS    X                indicator of this event
       DS    99XL4
NPFSHOOK_TABLE# EQU   100
NPFSIND_2 DS    AL1                  indicator
NPFSUNUSED_6 EQU    X'FE'            unused
NPFSUNLOAD      X'01'                unload after STOP-SS-COSMOS
NPFS#    EQU   *-NPFSHDR
```

Return Codes

SRC2	SRC1	MRC	Meaning
00	00	0000	no error detected
00	40	0001	COSMOS already active
00	40	0002	warning during task-selection
00	40	0003	error during macro processing; no term
00	20	0004	error during macro processing; term
00	20	0005	COSMOS-DMS error
00	40	0006	TLT error
00	20	0007	BTASK not activated