1 Preface

PROP-XT stands for **pr**ogrammed **op**erating with the e**xt**ended language resources of SDF-P.

1.1 Brief product description

PROP-XT is a subsystem in BS2000.
PROP-XT provides administration commands which are used to obtain various types of information on PROP-XT and on active administration processes. These commands can also be used to perform simple administration tasks without having to create administration procedures. Administration commands are entered via UCON.

PROP-XT also provides users with a programming interface, enabling them to automate BS2000 administration tasks quickly and easily using the SDF-P programming language. Administration procedures, containing PROP-XT system commands alongside SDF-P and BS2000 commands, are created in the BS2000 command language. These administration procedures run in administration processes. Using the UCON interface for programmed operating, these processes can manage administration entities.

1.2 Target group

This manual is intended for system administrators, system users, work process schedulers, and anyone involved in similar activities.

You must be familiar with the fundamentals of BS2000, and know how the system operates. If you wish to create your own administration procedures, you must also be familiar with the SDF-P programming language and the product JV (Jobvariable).

1.3 Summary of contents

The chapter “PROP-XT functionality” provides an overview of the PROP-XT mode of operation, as well as definitions of the terms required to understand PROP-XT.

“Administration interface” contains detailed descriptions of the PROP-XT administration commands.

“Introduction to programming with PROP-XT” describes the typical features of an administration procedure, explains how messages are processed using examples, and provides general programming information.

“Programming interface” contains detailed command descriptions and an example of a complete administration procedure, and describes a test system for testing administration procedures.

“User-defined commands” provides information on how to work with commands that you can define.

“Generation and startup” provides details on DSSM and PROP-XT generation, as well as syntax files and statements for starting and terminating PROP-XT.

“SYSTEM DUMP MANAGER - a PROP-XT application“ provides information on the SDM application, with which you can administer system dumps more easily.

“Messages” is a catalog of all PROP-XT messages sorted according to their hexadecimal codes.

The final part of the manual contains the glossary, a list of related publications, and an index.

1.4 Changes from the previous edition

PROP-XT V1.1A can be used in a network. Administration entities can be situated both on your own system and on remote systems.

Connections to UTM and DCAM applications can also be established via the PROP-XT interface. This means that administration entities can be of the OPERATING type as well as of the UTM or DCAM type.

Administration procedures can be exchanged at runtime without any loss of information.

The administration processes can run without interruption despite changeovers to summer or winter time and the use of local time.

New administration interface command:

SHOW-PROP-DATABASE (cf. page 54)

Extended administration interface commands:

CREATE-PROP-OBJECT-DESCRIPTION (cf. page 42)

MODIFY-PROP-PARAMETERS (cf. page 47)

SEND-TO-PROP-PROCESS (cf. page 52)

SHOW-PROP-PARAMETERS (cf. page 56)

Extended programming interface commands:

SEND-TO-PROP-ADMINISTRATOR (cf. page 88)

SEND-TO-PROP-OBJECT (cf. page 93)

SIMULATE-PROP-EVENT (cf. page 97)

START-PROP-EVENT-MONITORING (cf. page 101)

START-PROP-OBJECT-MONITORING (cf. page 110)

WAIT-FOR-PROP-EVENTS (cf. page 120)

1.5 Notational conventions

The following conventions are used in this manual:

- References to other publications are indicated in the text by an abbreviated title. The complete title, together with a brief description, is given in the “Related publications” section at the end of this manual.
- References to other sections within this manual contain the relevant page number and, where appropriate, the name of the chapter or section.
- References to topics described in other manuals contain the abbreviated title of the manual. The “Related publications” section contains the full title. You can then consult the index of the corresponding manual and locate the appropriate text section.
- <Variable names> in angle brackets should be replaced by actual names.
- System outputs (e.g. messages) and examples are printed in `typewriter text`. System inputs appear in **`bold typewriter text`**.
- **Bold** type is used to highlight important terms.

ℹ Symbol indicating particularly important information.

SDF metasyntax

Identification	Meaning	Examples
UPPERCASE LETTERS	Uppercase letters identify key words. Some key words begin with *.	**MODE** = **UNCH**ANGED **NAME** = ***ALL**
UPPERCASE **LETTERS** in bold	Bold uppercase letters identify guaranteed or proposed key word abbreviations.	**CHECK-VERS**ION = ***NO**
=	The equals sign links an operand name with the associated operand values.	**SCOPE** = **TEMP**ORARY
< >	Angle brackets indicate variables whose value set is defined by the data type and additional qualifications (see table 2).	**NAME** = ***NONE** / <name 1..8>
underscoring	Underscoring identifies the default value of the operand.	**REPL**ACE-**PROD**UCT = **YES** / **NO**
/	A slash is used to separate alternative operand values.	**MODE** = **UPD**ATE / **READ**
(...)	Parentheses are used to enclose operand values that indicate the start of a structure.	**TYPE** = **GROUP**(...)
indentation	Indentation indicates a dependency on the superordinate operand.	**OBJECT** = **DOMAIN**(...) **DOMAIN**(...) │ **NAME** = ***ALL**
│	A vertical line identifies the operands belonging to a structure. It marks the beginning and end of a structure. A structure can contain further structures. The number of vertical lines before an operand indicates the structure depth.	**SUP**PORT = **TAPE**(...) **TAPE(...)** │ **VOL**UME = ***ANY**(...) │ ***ANY**(...) │ │ ...
,	A comma is placed before additional operands of the same structure level.	,**SIZE** = **MIN**IMUM ,**FORM** = **UNGUID**ED

Table 1: Metasyntax (Section 1 of 2)

Identification	Meaning	Examples
list-poss(n):	A list can be formed from the operand values that appear after *list-poss*. If (n) is specified, the list can contain up to n elements. If the list contains more than one element, it must be enclosed in parentheses.	list-poss: **SAM** / **ISAM** list-poss(40): <structured-name 1..30> list-poss(256): ***OMF** / ***SYSLST**(...) / <full-filename 1..54>

Table 1: Metasyntax (Section 2 of 2)

Data types

Data type	Character set	Special features
alphanum-name	A…Z 0…9 $, #, @	
c-string	EBCDIC	Must be enclosed in single quotes; can be preceded by the letter C; single quotes within the c-string must be changed to double quotes
date	0…9 Structure indicator: hyphen	Input format: yyyy-mm-dd yyyy: year; either 2- or 4-position mm: month dd: day
full-filename	A…Z 0…9 $, #, @ hyphen period	Input format: $\text{:cat:\$user.}\left\{\begin{array}{l}\text{file}\\\text{file(no)}\\\text{group}\\\text{group}\left\{\begin{array}{l}\text{(*abs)}\\\text{(+rel)}\\\text{(-rel)}\end{array}\right\}\end{array}\right\}$

Table 2: Data types (Section 1 of 3)

Data type	Character set	Special features

full-filename (continued)		:cat:
		Optional specification of the catalog ID; character set restricted to A…Z and 0…9; max. 4 characters; must be enclosed in double quotes;
		default value is the catalog ID assigned to the user ID in the user catalog entry.
		$user.
		Optional specification of the user ID; character set A…Z, 0…9, $, #, @;
		max. 8 characters; must not begin with a digit; $ and periods must be specified;
		default value is your own user ID.
		$. (special case)
		Default system ID
		file
		Name of file or job variable; last character must not be a hyphen or a period; max. 41 characters; must contain at least one letter (A…Z).
		#file (special case)
		@file (special case)
		If the first character is # or @, this indicates temporary files and job variables depending on the system generation.
		file(no)
		Tape file name
		no: version number;
		character set A…Z, 0…9, $, #, @.
		Parentheses must be specified.
		group
		Name of a file generation group
		(character set same as “file”)

Table 2: Data types (Section 2 of 3)

Data type	Character set	Special features
full-filename (continued)		group $\left\{\begin{array}{l}\text{(*abs)}\\\text{(+rel)}\\\text{(-rel)}\end{array}\right\}$ (*abs) Absolute generation number (1-9999); * and parentheses must be specified. (+rel) (-rel) Relative generation number (0-99); sign and parentheses must be specified.
integer	0…9, +, -	+ or - can only appear as the first character.
name	A…Z 0…9 $, #, @	Must not begin with a digit.
structured-name	A…Z 0…9 $, #, @ hyphen	Alphanumeric character string which can be divided into several substrings separated by hyphens; first character: A…Z or $, #, @
time	0…9 Structure indicator: colon	Time specification Input format: $\left\{\begin{array}{l}\text{hh:mm:ss}\\\text{hh:mm}\\\text{hh}\end{array}\right\}$ $\left.\begin{array}{ll}\text{hh:} & \text{hours}\\\text{mm:} & \text{minutes}\\\text{ss:} & \text{seconds}\end{array}\right\}$ leading zeros can be omitted
x-string	Hexadecimal: 00…FF	Must be enclosed in double quotes; must be preceded by the letter X.

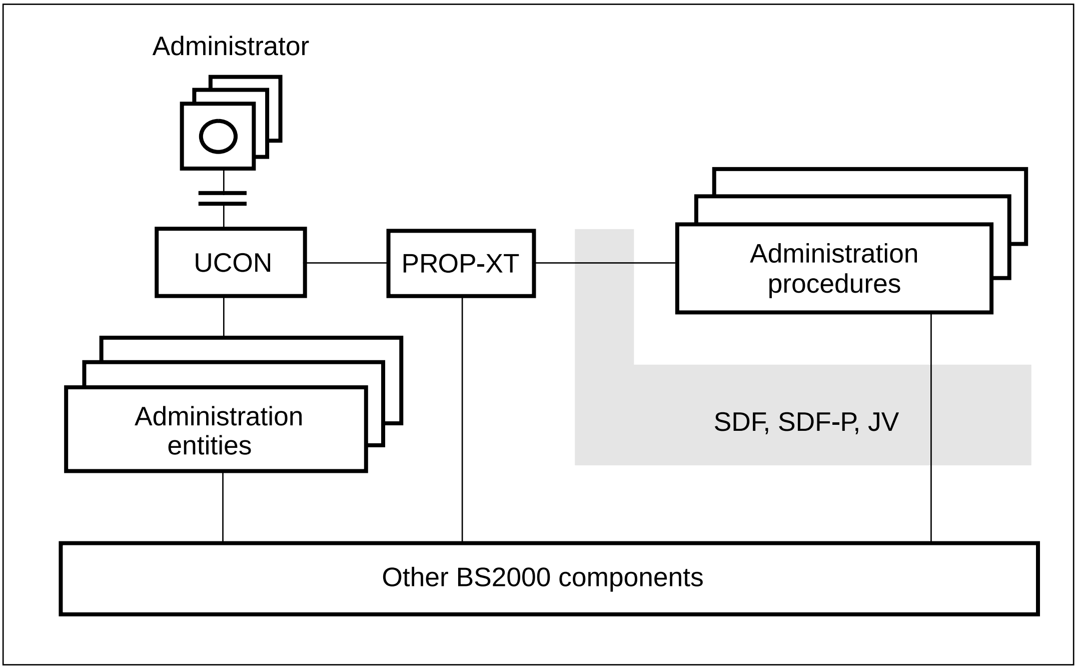
Table 2: Data types (Section 3 of 3)

2 PROP-XT functionality

This chapter provides an overview of the PROP-XT mode of operation and functions, as well as explaining the terms required to understand PROP-XT.

PROP-XT is a subsystem in BS2000, which allows for programmed operating using the SDF-P language resources. It requires BS2000/OSD-BC as of V1.0.

The diagram below is a simplified representation of the PROP-XT system environment. The terms and their interrelationships are explained in detail in the following sections.



You create administration procedures in the BS2000 command language, which contain the following commands:

- SDF-P commands (see the SDF-P manual)
- BS2000 commands (see the “User Commands” manual, Volume 1 and 2)
- PROP-XT system commands

Administration procedures run in administration processes.

The interface between PROP-XT and the administration procedures consists of:

- PROP-XT system commands
- SDF-P variables

An administration process can manage administration entities via PROP-XT and UCON (see the System Operator’s Guide).

The job types defined by UCON are available for communication with administration entities of the OPERATING type:

- command
  (input, results, additional information requests, additional information, termination)
- question
  (input, reply)
- message
  (input)

As a UCON application, PROP-XT processes $CONSOLE commands, sends and receives command-related messages, and thus assumes the role of a command server for the commands it activates.

PROP-XT system commands are used for signing administration processes and administration entities on and off, for defining and awaiting events, and for sending information to administrators and administration entities.

In addition to system commands, PROP-XT also provides administration commands. These administration commands are entered via UCON. They are used by the administrator at runtime to set up the database required by PROP-XT, request information on the entities known to PROP-XT (e.g. via administration processes), control PROP-XT, and automate simple administration tasks (reactions to defined events) via PROP-XT.

It is also possible to define your own commands using SDF-A (see the SDF-A manual). These user-defined commands are entered by the administrator and are passed by PROP-XT to the administration procedures for processing.

All communication between PROP-XT, the administration entities, and the administration processes takes place via UCON. UCON ensures that all messages are logged in the CONSLOG log file (see the System Administrator’s Guide or the Introductory Guide to Systems Support).

2.1 Administration processes

Administration processes are user processes running administration procedures.

Each administration process has a name which identifies the process uniquely throughout the system. An administration process specifies its name when it is signed on using the system command BEGIN-PROP-PROCESS.

Each administration process is a separate BS2000 task which runs under a particular user ID.

Administration processes are generally started using ENTER and run in batch mode. However, it is also possible to run them in interactive mode, which is particularly useful when developing administration procedures.

An administration process usually contains a complete subtask of automatic administration activities. Within this subtask, the administration process defines events dynamically (message events, job variable events, time events), and waits for the occurrence of these events. The definition of and waiting for events can be undone.

Signing administration processes on and off

In PROP-XT, administration processes are signed on as “running” or “for testing” using the system command BEGIN-PROP-PROCESS. This command is analyzed by SDF and passed to PROP-XT. PROP-XT checks the task authorization and takes over management of the administration process. If an administration process is already signed on with the same ID, PROP-XT rejects the command.

Administration processes are signed off using the END-PROP-PROCESS command. This command is also executed implicitly when a task terminates, if the process is not already signed off. When signing off, all buffered results are deleted and task-specific data areas are released.

2.2 Administration procedures

Administration procedures are BS2000 procedures that contain PROP-XT system commands.

An administration procedure is written in the BS2000 command language, and contains:

- SDF-P commands
- BS2000 commands (see the “Commands” manuals)
- PROP-XT system commands

The products SDF-P (see the SDF-P manual) and JV (Job Variables, see the JV manual) provide the language resources necessary for programming. Administration procedures can be controlled as desired using the SDF-P resources (CALL-PROCEDURE).

The section “Test system” on page 135ff contains a detailed description of how to test administration procedures.

2.3 Administration entities

Administration entities are objects which require administrator intervention and are managed in administration processes.

Administration entities can be situated both on your own system and on a remote system you can access. They can be of the OPERATING, UTM or DCAM type.

When PROP-XT manages entities of the UTM or DCAM type, this does not mean that the UTM/DCAM subsystems are administered, but only that there are connections to DCAM and UTM applications via the PROP-XT interface.

Setting up connections to administration entities

You have to define each administration entity for PROP-XT, providing in particular information on the type, authorization and addressing. The system command START-PROP-OBJECT-MONITORING is used for this.

You can specify the information on the type, authorization and addressing either directly in START-PROP-OBJECT-MONITORING or state in this command that the information is on file in the database. Information is stored in the database using the administration command CREATE-PROP-OBJECT-DESCRIPTION.

OPERATING type

PROP-XT sets up a connection to $CONSOLE for each administration entity of the OPERATING type defined with START-PROP-OBJECT-MONITORING. Thereafter, you can send data to this administration entity or receive information from it: questions, messages, additional information, answers to questions, console commands (which are permitted for the user in question) etc.
The number of possible simultaneous connections is limited in the BS2000, with the result that the number of entities that can be simultaneously managed in PROP-XT is also restricted.

DCAM type

PROP-XT sets up a connection to the specified DCAM application for each administration entity of the DCAM type defined with START-PROP-OBJECT-MONITORING.

Features of connections to administraton entities of the DCAM type:

- PROP-XT sets up a BCAM connection with an NEA protocol (i.e. there are no edit options). The administration entities must therefore be able to understand the NEA protocol.
- There are not formats, no support for virtual terminals, and no telegram communications.
- The priority is low.
- Messages are not combined into larger logical units. A message must not be longer than 1800 charcaters, or else it is truncated.
- The administration entity may only send printable characters, otherwise the characters cannot be edited in an SDF-P variable.

CMX or APS applications can also be treated by PROP-XT like administration entities of the DCAM type. However, these applications must not require any CMX- or APS-specific handling by PROP-XT at the IBCAM interface.

UTM type

PROP-XT sets up a connection to the specified UTM application for each administration entity of the UTM type defined with START-PROP-OBJECT-MONITORING.

For administration entities of the UTM type, work with a particular PROP application or work with programs in general must be permitted in the KDCDEF of the UTM application. This is done by making relevant specifications in the TPOOL or LTERM and PTERM statement. Messages are sent to the UTM application using SEND-TO-PROP-OBJECT. PROP-XT maps the messages at the BCAM level to the message level “Transaction” (BCAM_PARAM.LEV=’T’).

Multiplex connections are not supported.

Passwords

A password is not needed for connections to administration entities of the OPERATING type which are on a system which operates together with PROP-XT in an MSCF multiprocessor system of the type CCS (Closely Coupled System).
A password is required for all other connections of the OPERATING type.

In the case of entities of the DCAM/UTM type, the DCAM/UTM application decides whether a password is required.

The database information and thus the passwords are retained after PROP-XT is restarted if they are stored in a file (see the command MODIFY-PROP-PARAMETERS on page 47).

i A database file represents a certain safety risk! The passwords must be stored in the file in two-way encrypted form. This means that the encryption algorithm and the encryption password have to be known to decrypt the passwords.

For reasons of safety, you should therefore not use a database file and should only work with the database in the memory.

If you nevertheless wish to use a database file, you should restrict access to it. Using GUARDS, you can permit access to the file from PROP-XT only. For every definition in the database, you can also specify that only administration processes under specific user IDs may use the information (see the “SECOS“ manual).

Clearing down connections to administration entities

The connection is cleared down when an administration processes terminates itself, or signs off monitoring of the administration entity (STOP-PROP-OBJECT-MONITORING).

Communicating with administration entities

The system commands START-PROP-EVENT-MONITORING, WAIT-FOR-PROP-EVENTS, SEND-TO-PROP-OBJECT, and STOP-PROP-EVENT-MONITORING are used for communication between the administration processes and administration entities. The commands START-PROP-OBJECT-MONITORING and STOP-PROP-OBJECT-MONITORING are used to set up and clear down connections to administration entities.

2.4 Administrators

Administrators are all entities that manage administration entities, especially PROP-XT, and administration processes.

The administrator can therefore be one of the following:

- a system operator at a physical console
- a system operator at an OMNIS terminal
- an OMNIS-PROP application
- a PROP-XT administration process
- any authorized user process

See also the System Operator's Guide, and the manuals "OMNIS Administration and Programming" and "OMNIS-PROP".

Administration entities of the OPERATING type are managed via UCON. The behavior of administrators must comply with that defined by UCON. This behavior supports:

- job types (commands, questions, and messages)
- job attributes (e.g. job IDs)

Communicating with administration procedures

Corresponding system commands are provided at the programming interface for communication between administration procedures and administrators (START-PROP-EVENT-MONITORING, WAIT-FOR-PROP-EVENTS, SEND-TO-PROP-ADMINISTRATOR).

An administrator establishes a connection to an administration process by issuing a command that is executed by this process. An administrator definition is not required in the administration procedure. PROP-XT manages administrators insofar as it ensures that job-related messages, identified by the job ID for messages (REFERENCE-NAME) within command processing (result, additional information request, termination), reach the correct administrator.

If an administration process initiates the connection to an administrator (e.g. to output a message or question), the UCON administrator is selected.

Signing on administrators

Administrators are not signed on directly to PROP-XT. They are not managed by PROP-XT, rather by UCON. PROP-XT cannot distinguish between individual administrators.

The connection to UCON, which is used by administrators to manage administration entities and processes, is established immediately after PROP-XT is started.

2.5 Jobs

In the context of PROP-XT, jobs are the messages and message sequences described in the following.

i Only administration entities of the OPERATING type are affected. Jobs are not supported for administration entities of the UTM or DCAM type.

Messages and message sequences are exchanged between administrators and administration procedures, or between administration procedures and administration entities of the OPERATING type.

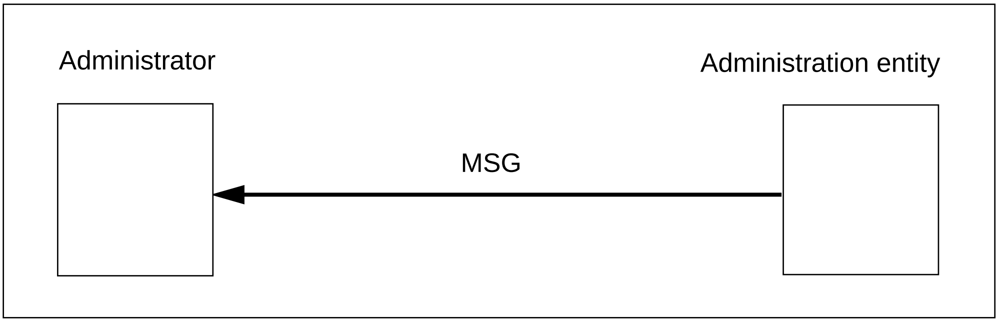
The behavior of these messages must conform with that defined by UCON. This behavior supports:

- job types (commands, questions, and messages
- job attributes (e.g. job IDs)

The diagrams below illustrate how message sequences of the various job types supported by UCON are processed.

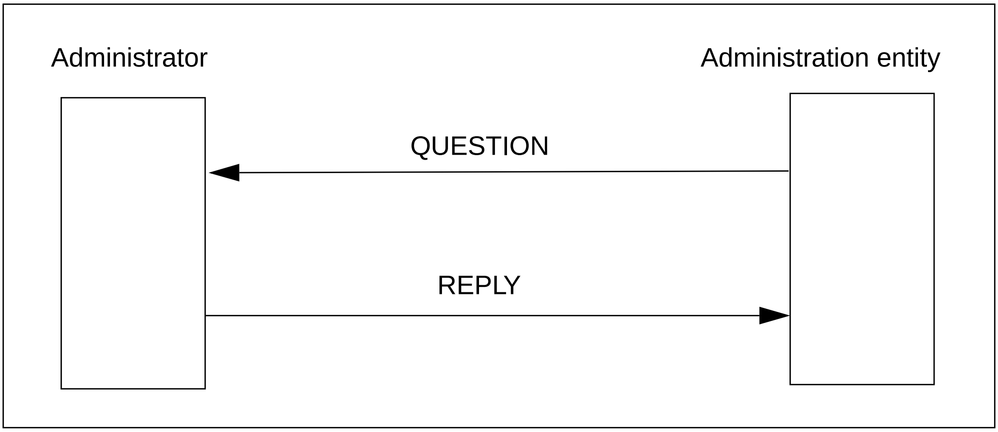
Messages

The message job type corresponds to a single message.



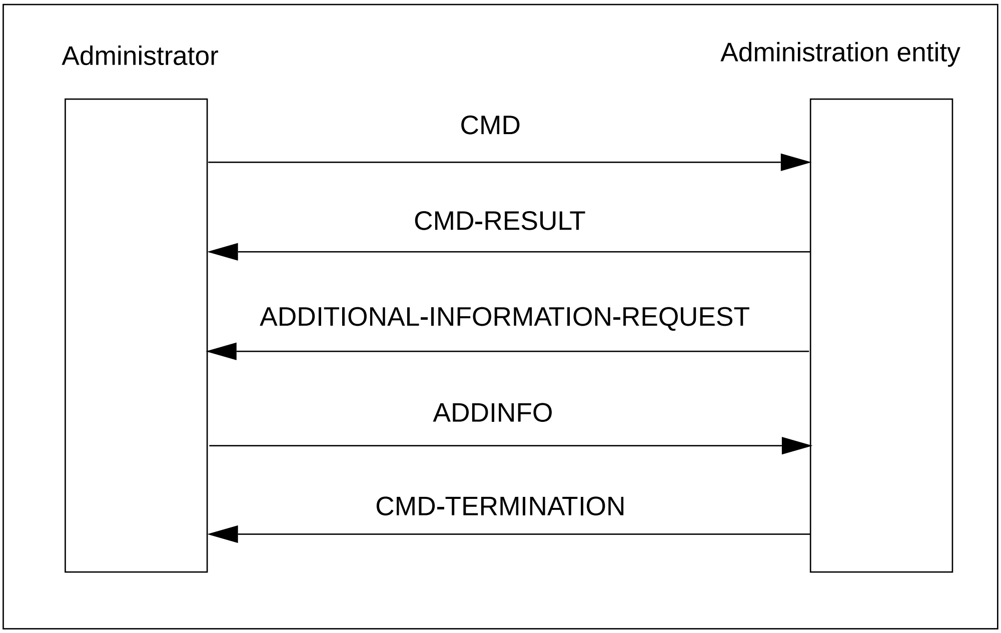
Questions

A message followed by a single reaction corresponds to the question job type.



Commands

A multipart message sequence corresponds to the command job type. With a command, all the messages in the sequence are identified as such (command bracketing).
When message sequences are processed in PROP-XT, the user can specifically wait for results (with WAIT-FOR-PROP-EVENTS) or query them (with CHECK-PROP-EVENT-OCCURRENCE).



2.6 Events

An administration process is activated when one of a set of predefined events occurs. The administration process must define which events activate it.

Definition of events

Administration processes must inform PROP-XT of the events they process. This set of events is dynamic, i.e. events can be added or removed at any time.

The set of events is managed using the commands START-PROP-EVENT-MONITORING and STOP-PROP-EVENT-MONITORING.
An implicit START-PROP-EVENT-MONITORING for the results is triggered when a command or question is sent:

- (SEND-TO-PROP-OBJECT,TYPE=CMD(IMPLICIT-EVENTS=*ALL/*CMD-TERMINATION),DATA=...,REFERENCE-NAME=...)
- (SEND-TO-PROP-ADMINISTRATOR,TYPE=*QUESTION,DATA=...,REFERENCE-NAME=...)

Event types

A distinction is made between the following event types:

- message events
- command events
- job variable events
- time events
- job events

Message events

A message event occurs when PROP-XT receives a message. The message event is sent to all administration processes which are signed on for this event, and which are authorized to process the message.

> Messages from UTM or DCAM applications can only be received as message events.

A message event is defined using the START-PROP-EVENT-MONITORING, with SELECT-EVENT=*FROM-OBJECT (see page 101).

Command events

Command events occur when an administrator enters a user-defined command. An administration process must specify which commands it processes.

> A command event is defined using the command START-PROP-EVENT-MONITORING, with SELECT-EVENT=*FROM-ADMINISTRATOR (see START-PROP-EVENT-MONITORING (see page 101)).

All user-defined commands must be defined using SDF-A. The syntax definition must be contained in the syntax file assigned to the user ID.

Job variable events

An event occurs when a particular job variable is modified. Job variable events are defined explicitly. It is possible to simultaneously wait at a common wait point for any defined event and for the modification of job variables.

> A job variable event is defined using the command START-PROP-EVENT-MONITORING, with SELECT-EVENT=*JV-MODIFICATION (see page 101).

Time events

With PROP-XT, users can specify a start time and, in some cases, an interval for time events. The time event occurs for the first time at the defined start time. In the case of an interval, it occurs each time the interval expires.

> A time event is defined using the command START-PROP-EVENT-MONITORING, with SELECT-EVENT=*TIMER (see START-PROP-EVENT-MONITORING page 101).

You can choose whether to work with Universal Coordinated Time (ZONE=UTC) or local time (ZONE=LOCAL-TIME). If local time is used, the SEASON operand ensures uninterrupted operation even during a change from summer time to winter time and vice versa. SEASON is only evaluated in the time change phase. PROP-XT then calculates the necessary replacement times in hours lost or gained (double).

PROP-XT does not provide a direct interface for more complicated date events (e.g. first working day of each month). You can define such complex date events outside PROP-XT. It is possible to transfer this type of date event to PROP-XT in the form of a job variable event. For example, AVAS can be used to start a job in which a particular job variable is set (see the AVAS manual).

Job events

PROP-XT automatically generates job-related events when processing message sequences with question/reply or command/result/additional information request/additional information/termination (see page 19). The name of a job event is defined by

REFERENCE-NAME (commands SEND-TO-PROP-ADMINISTRATOR and SEND-TO-PROP-OBJECT). It is also possible to wait for job events using WAIT-FOR PROP-EVENTS.

Identifying events

Every user-defined event is identified by EVENT-NAME, which must be specified in the START-PROP-EVENT-MONITORING command. This name must be unique within an administration process. However, different administration processes can have the same name for different events.

If an event is defined twice with different event IDs, PROP-XT regards this as two different events.

Incoming messages are assigned to all events that identify these messages, and may thus be sent several times (once per ID) using WAIT-FOR-PROP-EVENTS.

Commands are always forwarded to one executor only. When forwarding commands to the current command executor, PROP-XT behaves in the same way as UCON (see the System Operator’s Guide).

Event groups

When defining events with START-PROP-EVENT-MONITORING, each event can be assigned to an event group. The event group itself is defined the first time it occurs using the START-PROP-EVENT-MONITORING command. It provides a simple means of simultaneously addressing several events in the PROP-XT commands WAIT-FOR-PROP-EVENTS, CHECK-PROP-EVENT-OCCURRENCE, and STOP-PROP-EVENT-MONITORING.

Waiting for events

WAIT-FOR-PROP-EVENTS is used to wait for events in an administration process. The events must be defined in advance, either explicitly with START-PROP-EVENT-MONITORING, or implicitly by sending a command or question with SEND-TO-PROP-OBJECT or SEND-TO-PROP-ADMINISTRATOR (see the section “Definition of events” on page 21.

The events to be awaited can be selected using various criteria:

- event ID
- event group ID
- sender
- job ID that was specified in the command SEND-TO-PROP-OBJECT or SEND-TO-PROP-ADMINISTRATOR

The wait time can be restricted by means of the TIME-LIMIT operand.

Managing events

Events are buffered from the time they are defined by PROP-XT. The administration procedure can define whether all events are to be buffered, or only the last (chronological) event belonging to an event definition. An administration process can obtain information on the buffered events using CHECK-PROP-EVENT-OCCURRENCE. The operands of the CHECK-PROP-EVENT-OCCURRENCE command allow you to query specific events. However, the events are transferred only with the WAIT-FOR-PROP-EVENTS command.

Dynamic definition of events

After an administration process is signed on to PROP-XT, events can be defined in any order using START-PROP-EVENT-MONITORING and deleted using STOP-PROP-EVENT-MONITORING.

When an event occurs, it is buffered by PROP-XT for the administration procedure until it is retrieved.

The administration procedure is responsible for ensuring that requested events are retrieved, and that buffer overflows are avoided. Buffer overflows may result in loss of information.
The CHECK-PROP-EVENT-OCCURRENCE command is used to obtain information on buffered events, but does not remove events from the buffer.

Central event handling

WAIT-FOR-PROP-EVENTS is a programming option that enables you to wait at a central waiting point. All events signed on in the administration procedure are retrieved and processed in the order in which they occur. This programming method prevents buffer overflows.

Logically linked event sequences

In addition to the option of waiting for events at a central location, PROP-XT also makes it possible to wait for a single event or for one of a set of events. This allows for simple, problem-related programming.

Sending commands to an administration entity

If an administration process sends a command to an administration entity, the process can wait for command results (WAIT-FOR-PROP-EVENTS). Depending on command processing, the result may be a request for additional information. In all cases, a termination acknowledgment is sent to the administration process. Depending on the command, this termination acknowledgment may signify actual command termination, or merely the transfer of a command.

Working with job variables

A job variable value can be defined as an event using START-PROP-EVENT-MONITORING. The administration process can thus wait simultaneously for messages, time events, or job variable values using WAIT-FOR-PROP-EVENTS.

2.7 Commands

The PROP-XT commands can be divided into three classes:

Administration commands

These are commands used to control PROP-XT, obtain information, and define and manage standard tasks. They are processed by PROP-XT itself.

– commands for obtaining information

```
SHOW-PROP-DATABASE
SHOW-PROP-PARAMETERS
SHOW-PROP-PROCESS-STATUS
SHOW-PROP-STD-ACTION
```

– commands for controlling PROP-XT and administration processes

```
MODIFY-PROP-PARAMETERS
SEND-TO-PROP-PROCESS
```

– commands for performing simple administration tasks

```
ADD-PROP-STD-ACTION
REMOVE-PROP-STD-ACTION
```

– commands for describing administration entities

```
CREATE-PROP-OBJECT-DESCRIPTION
DELETE-PROP-OBJECT-DESCRIPTION
```

System commands

The system commands, i.e. programming interface commands, are used to program administration procedures.

- commands for signing administration processes and administration entities on and off

  BEGIN-PROP-PROCESS
  END-PROP-PROCESS
  START-PROP-OBJECT-MONITORING
  STOP-PROP-OBJECT-MONITORING

- commands for defining and waiting for events

  START-PROP-EVENT-MONITORING
  CHECK-PROP-EVENT-OCCURRENCE
  WAIT-FOR-PROP-EVENTS
  STOP-PROP-EVENT-MONITORING

- commands for sending information to administrators and administration entities

  SEND-TO-PROP-ADMINISTRATOR
  END-TO-PROP-OBJECT

- command for testing administration procedures

  SIMULATE-PROP-EVENT

User-defined commands

These can use SDF-A to define commands which are signed on to PROP-XT by means of events using START-PROP-EVENT-MONITORING.

If you enter this type of command, it is passed to the administration procedure at the central wait point (WAIT-FOR-PROP-EVENTS).

Like administration commands, user-defined commands are operator special commands.

2.8 Managing administration entities and processes

Managing administration entities using administration processes

Normally, administration entities are managed by administration processes. It is possible to define automatic responses to particular events using PROP-XT system commands in administration procedures, which run in administration processes.

Managing administration entities using administration commands

Administration commands enable you to perform simple administration tasks without having to create administration procedures.

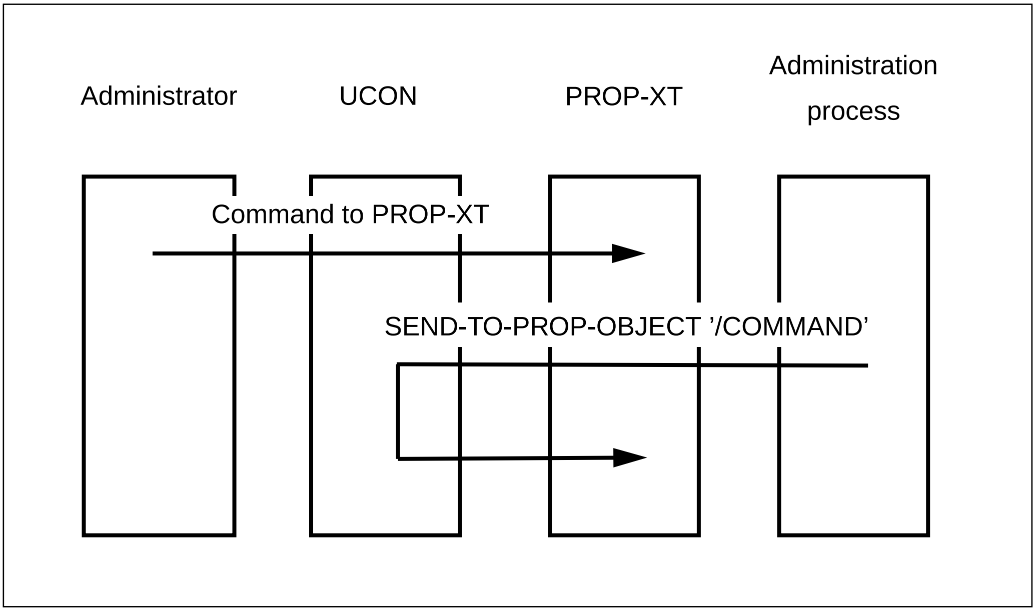
In a string of ADD-PROP-STD-ACTION commands, each command describes an event EVENT NAME relevant for administration (e.g. a console message), and the desired response ACTION (e.g. a console command).
The event EVENT-NAME is always a message that originates from the administration entity STANDARD-ACTION-AO. This entity must be defined in advance using CREATE-PROP-OBJECT-DESCRIPTION.
It is possible to define the responses of various administration entities, all of which must be made known using CREATE-PROP-OBJECT-DESCRIPTION.

Managing the PROP-XT administration entity

The administration interface commands, which are entered by the administrator, are used to manage PROP-XT.

PROP-XT is also a subsystem in BS2000, which can be managed using administration processes. In this context, PROP-XT is a system component and can be addressed as an administration entity of type UCON.

There is no direct interface for managing PROP-XT using administration processes. This is only possible using operator special commands via UCON, as shown in the following diagram:



Managing administration processes

PROP-XT does not provide special administration commands for managing administration processes.

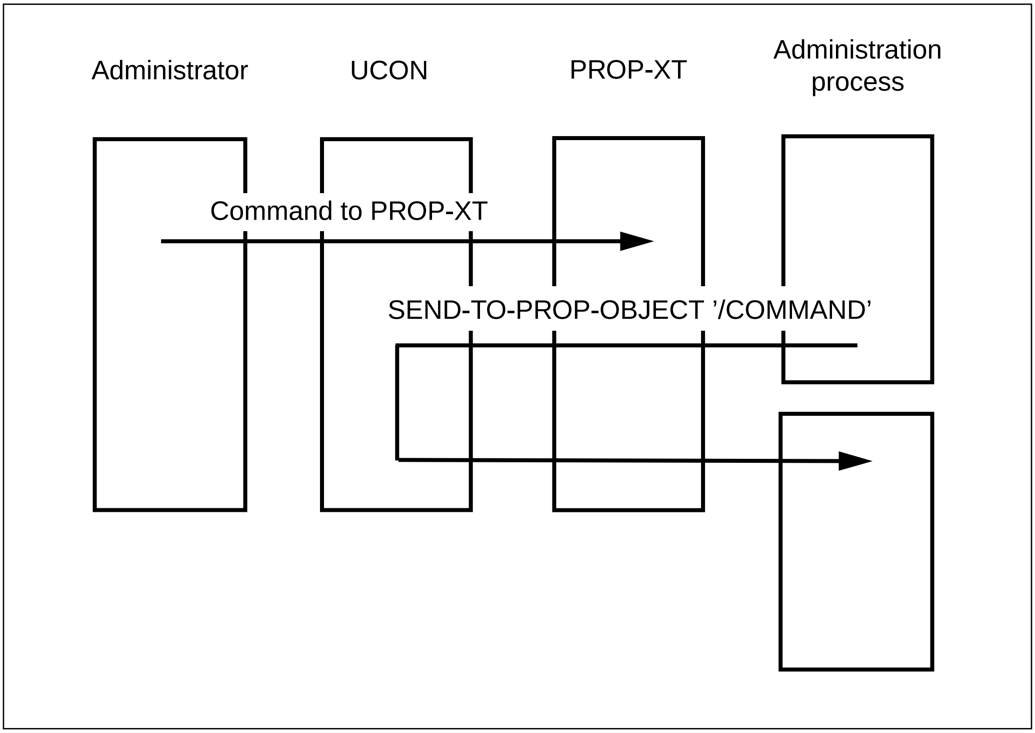
Administration processes can be started and terminated using BS2000 commands (ENTER-PROCEDURE, CALL-PROCEDURE, CANCEL, see the manual “Commands”, Volume 1).

However, administrators can intervene in administration processes by means of user-defined commands.

Administration processes as administration entities

Administration processes are also administration entities, and can therefore be managed as such. For this purpose, they must be capable of processing user-defined commands. These commands are entered by administrators, and are passed to the administration processes by PROP-XT.

There is no direct interface for managing administration processes using other administration processes. Administration processes can only be managed via UCON. In this case, the administration process to be managed is regarded as a UCON component. User-defined commands are operator special commands. The diagram below illustrates command execution:



2.9 Exchange of administration procedures

An administrator can replace an administration procedure with another administration procedure during runtime.

The administrator enters the command
SEND-TO-PROP-PROCESS TYPE=EXCHANGE-REQUEST(NEW-PROCEDURE=...) (see page 52). The active administration process that is addressed then starts the new administration procedure using the SDF-P command INCLUDE-PROCEDURE, i.e. the old administration procedure calls the new administration procedure as a subprocedure.

No events are lost in this process, nor are any processes or programs aborted and/or restarted. All activities are continued as before:

- The new administration procedure (Ap-new) continues to run in the same environment as the old one (Ap-old) - this affects SYSFILE assignments, user switches, file links, authorizations, connections etc.
- Ap-new takes over administration entities and events from Ap-old without any problems - Ap-new answers questions directed to Ap-old and receives the results of commands issued by Ap-old.
- Ap-new has read and write access to all variables of Ap-old.

Programming information

In Ap-old, EXCHANGE-REQUEST must be defined as an event using START-PROP-EVENT-MONITORING and recognized with WAIT-FOR-PROP-EVENT.

In Ap-new, you should not carry out duplicate initialization and termination, nor should you declare the variables again. If you do, the old values will no longer be visible. Exception: The variable SYSPOP also has to be declared in Ap-new for PROP-XT to be able to assign the variable.
You are recommended to query in the procedure whether you are performing a first-time start or not.

BEGIN-PROP-PROCESS, START-PROP-OBJECT-MONITORING and START-PROP-EVENT-MONITORING with existing names (i.e. which exist in both Ap-old and in Ap-new) have to be intercepted or skipped with IF-CMD-ERROR and IF-BLOCK-ERROR.

Ap-new has to declare RESULT-VARIABLE (with SEND-TO-PROP-ADMINISTRATOR, WAIT-FOR-PROP-EVENTS etc.) again.

Make sure that Ap-new branches, or can branch, to the correct position for exchange and, finally, that Ap-old is also terminated when Ap-new is concluded.

Example

The procedure AP.EXP.02 is to control the administration process AP-EXP presently controlled by the old procedure AP.EXP.01.
AP.EXP.02 must also be able to withstand an initial start and a restart. In addition, AP.EXP.02 should able to be exchanged with a new peocedure AP.EXP.03 during runtime.

Administration procedure AP.EXP.02:

```
/SET-PROC-OPTIONS.....

/   DECL-VAR SYSPOP,TYPE=STRUC
/IF  IS-DECLARED('RESTART-POINT',SCOPE=TASK) = FALSE
*   Initial start: Variables have not yet been declared
/   DECL-VAR RESTART-POINT
/   DECL-VAR
   .
   .
   .
   .
/ELSE      .
/   WRITE-TEXT TEXT=' Exchange of AP-EXP'
/   IF RESTART-POINT = FIRST-WAIT
/     GOTO LABEL=FIRST-WAIT
/   ELSE-IF RESTART-POINT = SECOND-WAIT
/     GOTO LABEL=SECOND-WAIT
/            .
/            .
/            .
/            .
/   END-IF
/END-IF

/BEGIN-BLOCK
* this block is executed only in the case of an initial start
/BEGIN-PROP-PROC  AP-EXP
/START-PROP-OBJECT-MONITORING  01
/SEND-T-P-O  01  DATA='REQUEST-OPERATOR-ROLE   ',REF=endquit
/WAIT-F-P-E EVENT-NAME=endquit
   .
   .
/START-PROP-EVENT-MONITORING EVENT-NAME=EXCHANGE,
     SELECT-EVENT=FROM-ADMINISTRATOR(TYP=EXCHANGE-REQUEST)

/START-PROP-EVENT-MONITORING
/START-PROP-EVENT-MONITORING

/END-BLOCK
   .
   .
```

```
/FIRST-WAIT:
/WAIT-F-P-E EVENT-NAME=(xyz,EXCHANGE)
/IF SYSPOP.EVENT-NAME = EXCHANGE
/    RESTART-POINT = 'FIRST-WAIT'
/    GOTO LABEL= EXCHANGE-LABEL
/ELSE-IF
   .
   .
   .
   .
   .
/SECOND-WAIT:
/WAIT-F-P-E EVENT-NAME=*ALL
/IF SYSPOP.EVENT-NAME = EXCHANGE
/    RESTART-POINT = 'SECOND-WAIT'
/    GOTO LABEL= EXCHANGE-LABEL
/ELSE-IF
   .
   .
   .
   .
/END-IF

/EXCHANGE-LABEL:
/INCLUDE-PROCEDURE &(SYSPOP.NEW-PROCEDURE)
/EXIT-PROC
```

2.10 Authorization checks

The authorization checks are used to verify:

- whether the administration process can execute PROP-XT system commands
- whether an administration process can manage a particular administration entity
- whether an administrator can manage PROP-XT
- whether an administrator can manage an administration process

PROP-XT is fully integrated in the BS2000 authorization check facility. UCON and PROP-XT cooperate to ensure that the user concept appears consistent.

The PROP-ADMINISTRATION privilege is required to execute PROP-XT commands. This privilege is automatically granted to the system administrator ID, but can also be assigned to another user ID if you are working with SECOS (optional) (see the SECOS manual).

Checking whether an administration process can execute PROP-XT system commands

A task can only be initiated as an administration process with BEGIN-PROP-PROCESS if the user ID under which it is running has the PROP-ADMINISTRATION privilege.
Thus, if you are working without SECOS, administration processes must run under the system administrator ID.

Checking whether an administration process can manage an administration entity

UCON regards PROP-XT and the administration processes it manages as various logical consoles. A number of administration processes run under different user IDs, and have different privileges.

The privileges of the individual administration processes are transferred to UCON (transparently to PROP-XT), where the authorization check is performed.

PROP-XT itself merely acts as a command server for PROP-XT administration commands, and has no other privileges.

Checking whether an administrator can manage PROP-XT

This is carried out by the UCON authorization check facility (see the System Operator’s Guide).

Each PROP-XT administration command has an authorization code (by default, R). Any administrator with this code can manage PROP-XT.

Each PROP-XT message has a routing code (by default, R). Any administrator with this code receives PROP-XT messages.

In addition, the PROP-ADMINISTRATION privilege is required by any administrator that is managed by UCON by means of its user ID, in order to enter PROP-XT administration commands.

Checking whether an administrator can manage an administration process

This is carried out by the UCON authorization check facility. Each user-defined command has an authorization code (by default, R). Any administrator with this code can manage administration processes.

3 Administration interface

The administration interface commands (or administration commands for short) are entered on the console, and are used to control PROP-XT, to obtain information, and to define and manage simple standard tasks.

3.1 Operation

The UCON authorization check facility verifies whether or not an administrator is authorized to manage PROP-XT: any administrator with the authorization code for PROP-XT administration commands (by default, R) can manage PROP-XT. In addition, the PROP-ADMINISTRATION privilege is required by any administrator managed by UCON, in order to enter PROP-XT administration commands.

Error information is output in message NBR0740 to the console from which the command was issued. The message contains the command return codes in the inserts MAINCODE, SUBCODE1, and SUBCODE2 (see table “General meanings of error classes” on page 79). Some administration commands also output runtime messages.

3.2 Administration commands

The notational conventions used in the following command descriptions are explained in the section “Notational conventions” on page 5ff.

ADD-PROP-STD-ACTION

This command is used to define a standard action STD-ACTION. A standard action in the context of PROP-XT is an automatic response to an event that is relevant for administration, without the need for an administration process.

Thus, you can perform simple administration tasks directly from PROP-XT without having to create an administration procedure. In a sequence of ADD-PROP-STD-ACTION commands, each command describes an event and the desired reaction. An EVENT-NAME, which must be unique throughout PROP-XT, is used to identify each assignment of an event to a response. This name enables you to refer to previous commands.

The event is always a message that originates from the STANDARD-ACTION-AO administration entity. This entity must be described in advance using CREATE-PROP-OBJECT-DESCRIPTION (see page 42ff). If the reaction involves sending a command to another administration entity, this entity must likewise be defined beforehand using CREATE-PROP-OBJECT-DESCRIPTION.

Information on an administration entity is transferred the first time a STD-ACTION refers to the administration entity. If no further STD-ACTIONs refer to a particular administration entity, the connection to this entity is cleared down and the administration entity is removed from the administration facility. If an administration entity description is modified, the changes take effect the next time the administration entity is incorporated in the administration of STD-ACTIONs.

```
ADD-PROP-STD-ACTION

 EVENT-NAME = <structured-name 1..30>
,EVENT-DATA = *SYSTEM-MSG (...) / *FREE-STRING(...)
     *SYSTEM-MSG(...)
          MSG-ID = <name 7..7>
         ,INSERT-0 = *NO / <c-string 1..180>
         ,INSERT-1 = *NO / <c-string 1..180>
         ,INSERT-2 = *NO / <c-string 1..180>
         ,INSERT-3 = *NO / <c-string 1..180>
         ,INSERT-4 = *NO / <c-string 1..180>
         ,INSERT-5 = *NO / <c-string 1..180>
         ,INSERT-6 = *NO / <c-string 1..180>
         ,INSERT-7 = *NO / <c-string 1..180>
         ,INSERT-8 = *NO / <c-string 1..180>
         ,INSERT-9 = *NO / <c-string 1..180>
         ,INSERT-10 = *NO / <c-string 1..180>
         ,INSERT-11 = *NO / <c-string 1..180>
         ,INSERT-12 = *NO / <c-string 1..180>
         ,INSERT-13 = *NO / <c-string 1..180>
         ,INSERT-14 = *NO / <c-string 1..180>
     *FREE-STRING(...)
          POSITION = 1 / <integer 1..1800>
         ,STRING = <c-string 1..1800>
,ACTION = *CMD (...) / *REPLY(...)
     *CMD(...)
          OBJECT-NAME = <structured-name 1..30>
         ,DATA = <c-string 1..500>
     *REPLY(...)
          DATA = <c-string 1..500>
,REPLACE-OLD-ACTION = *NO / *YES
```

EVENT-NAME = <structured-name 1..30>

This operand specifies an event ID. This name must be unique within the STD-ACTIONs.

EVENT-DATA =
This can be a comparison string, or an MSG7 message number and inserts. A message results in an event if it points to the specified comparison string, or if it has the specified message code and identical inserts to those entered here.

EVENT-DATA = *SYSTEM-MSG
A message code and the defined inserts are used for the comparison.

MSG-ID = <name 7..7>
Message code.

INSERT-i = *NO / <c-string 1..180>
INSERT-i identifies the message insert with the logical number i (where 0 ≤ i ≤ 14). The value *NO means that the value of insert i is not used for comparison.
If a character string is specified here, this must be a defined insert value, because only then will it be identified as an event.

EVENT-DATA = *FREE-STRING (...)
A free text string is used for the comparison.

POSITION = 1 / <integer 1..1800>
Position of the comparison string. The default value is the start of the text (position 1).

STRING = <c-string 1..1800>
Value of the comparison string.

ACTION =
Commands or replies to questions are sent to administration entities. This operand specifies the type of reaction to the event.

ACTION = *CMD (...)
Commands are sent to administration entities.

OBJECT-NAME = <structured-name 1..30>
Name of the administration entity to which the command is to be sent. The entity must be defined in the database if this is the first time this entity is referenced. If the administration entity was addressed in a previous STD-ACTION, the administration entity description is not accessed.

DATA = <c-string 1..500>
Character string defining the reaction to the event specified in the command.

ACTION = *REPLY (...)
Replies to questions are sent to administration entities.

DATA = <c-string 1..500>
Character string defining the reaction to the event specified in the command.

REPLACE-OLD-ACTION =
This operand is provided for dynamic modification of the generation at runtime.

REPLACE-OLD-ACTION = *NO
An entry is made only if PROP-XT cannot find an existing STD-ACTION with this name.

REPLACE-OLD-ACTION = *YES
If a STD-ACTION with the specified EVENT-NAME exists, it is modified by the parameters specified in the command.

Return codes when the command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP0902	Internal error	STD-ACTION could not be accepted.
POP0107	Semantic errors	Administration entity not of type UCON.
POP0108		No more administration processes/entities allowed.
POP090E		No response administration entity definition.
POP090F		No administration entity definition for STD-ACTION administration entity.
POP0901		STD-ACTION already exists. Command not executed.
POP0910		Specified type of STD-ACTION administration entity not supported.

Table 3: Return codes of the ADD-PROP-STD-ACTION command

CREATE-PROP-OBJECT-DESCRIPTION

This command is used to describe an administration entity and store the data in the database.

It assigns a name to the administration entity, and contains all the information required to access the entity and perform a correct authorization check. All administration entities used by STD-ACTION must be defined using CREATE-PROP-OBJECT-DESCRIPTION. This includes the STANDARD-ACTION-AO entity and all entities addressed in the ADD-PROP-STD-ACTION command (see page 38ff).

```
CREATE-PROP-OBJECT-DESCRIPTION

 OBJECT-NAME = STANDARD-ACTION-AO / <structured-name 1..30>
,OBJECT = *OPERATING (...) / *UTM-APPLICATION (...) / *DCAM-APPLICATION (...)
    *OPERATING(...)
      |   USER-IDENTIFICATION = *OWN / <name 1..8>
      |  ,PASSWORD = *OWN / *NO / <c-string 1..8> / <x-string 1..16>
      |  ,PROCESSOR-NAME = *LOCAL / <name 1..8>
      |  ,OPERATOR-ROLE = *NO / <name 1..8>
    *UTM-APPLICATION(...)
      |   APPLICATION-NAME = <name 1..8>
      |  ,PASSWORD = *NO /  <c-string 1..4> / <x-string 1..8>
      |  ,PROCESSOR-NAME = *LOCAL / <name 1..8>
      |  ,OWN-APPLICATION = *BY-PROP / <name 1..8>
    *DCAM-APPLICATION(...)
      |   APPLICATION-NAME = <name 1..8>
      |  ,PASSWORD = *NO /  <c-string 1..4> / <x-string 1..8>
      |  ,LOGON-MSG = *NO /  <c-string 1..80> / <x-string 1..160>
      |  ,PROCESSOR-NAME = *LOCAL / <name 1..8>
      |  ,OWN-APPLICATION = *BY-PROP / <name 1..8>
,REPLACE-OLD-OBJECT = *NO / *YES
,USER-ACCESS = *ALL-USERS / list-poss(20):<name  1..8>
```

OBJECT-NAME = STANDARD-ACTION-AO / <structured-name 1..30>

Name of the administration entity.

OBJECT = *OPERATING (...)

The administration entity is of the *OPERATING type.

USER-IDENTIFICATION = *OWN / <name 1..8>
User ID for the administration entity. The specification *OWN causes a connection to be established only if the administration entity is on the local processor or on a processor which works with PROP-XT in an MSCF network of the type CCS. This remote processor must have at least BS2000 version 11.2.
The specification refers to the ID during processing by START-PROP-OBJECT-MONITORING.

PASSWORD = *OWN / *NO / <c-string 1..8> / <x-string 1..16>
Passwort for setting up a connection to the administration entity.

The specification *OWN causes a connection to be established only if the administration entity is on the local processor or on a processor which works with PROP-XT in an MSCF network of the type CCS. USER-IDENTIFICATION=*OWN must also be specified.

The specification *NO means that the user ID is not password protected.

In all other cases, the correct password for the user ID has to be specified.

PROCESSOR-NAME = *LOCAL / <name 1..8>
Name of the processor on which the administration entity is situated. The specification *LOCAL means that the administration entity is on the local processor.

OPERATOR-ROLE = *NO / <name 1..8>
If an OPERATOR-ROLE is specified, it is requested by PROP-XT after the connection has been set up.

OBJECT = *UTM-APPLICATION (...)
The administration entity is a UTM application.

APPLICATION-NAME = <name 1..8>
Name of the UTM application.

PASSWORD = *NO / <c-string 1..4> / <x-string 1..8>
Connection password for the UTM application.

PROCESSOR-NAME = *LOCAL / <name 1..8>
Name of the processor on which the UTM application runs. The specification *LOCAL means that the administration entity is on a local processor.

OWN-APPLICATION = *BY-PROP / <name 1..8>
Specifies the auxiliary application with which PROP-XT sets up the connection to the UTM application.

When *BY-PROP is specified, PROP-XT assigns a unique, PROP-XT-wide name for the auxiliary application. Since this name is not known when the KDCDEF for the UTM application is generated, the UTM administrator must permit connections in general to programs by specifying PTYP=APPLI in the TPOOL statement.

The name for OWN-APPLICATION must not start with the reserved character $; it must be unique. <name 1..8> must be specified in the PTERM statement for the KDCDEF of the UTM application.

OBJECT = *DCAM-APPLICATION (...)
The administration entity is a DCAM application.

APPLICATION-NAME = <name 1..8>
Name of the DCAM application.

PASSWORD = *NO / <c-string 1..4> / <x-string 1..8>
Connection password for the DCAM-Anwendung.

LOGON-MSG = *NO / <c-string 1..80> / <x-string 1..160>
Connection message transferred when the connection is set up.

PROCESSOR-NAME = *LOCAL / <name 1..8>
Name of the processor on which the DCAM application runs. The specification *LOCAL means that the administration entity is on a local processor.

OWN-APPLICATION = *BY-PROP / <name 1..8>
Specifies the auxiliary application with which PROP-XT sets up the connection to the DCAM application. The name for OWN-APPLICATION must not start with the reserved character $; it must be unique.
When *BY-PROP is specified, PROP-XT assigns a unique, PROP-XT-wide name for the auxiliary application.
The DCAM application can recognize an administration procedure by means of the name of OWN-APPLICATION.

REPLACE-OLD-OBJECT =
This operand is provided for dynamic modification of the generation at runtime.

REPLACE-OLD-OBJECT = *NO
The description is transferred only if an administration entity with the specified name does not already exist.

REPLACE-OLD-OBJECT = *YES
If an administration entity with the specified name exists, it is overwritten by the definition specified here.

USER-ACCESS = *ALL-USERS / list-poss(20):<name 1..8>
Only the STD-ACTION may always use the description in the database. In addition, only the administration procedures which run under one of the user IDs specified here have access.

Return codes when the command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP0908	Internal error	Administration entity could not be accepted.
POP0100	Semantic errors	Administration entity already in database.
POP0107		Administration entity not of type UCON.

Table 4: Return values of the CREATE-PROP-OBJECT-DESCRIPTION command

Example

To define the STANDARD-ACTION-AO administration entity:

```
/CREATE-PROP-OBJECT-DESCRIPTION
    OBJECT-NAME=STANDARD-ACTION-AO,
    OBJECT=*OPERATING
        (USER-IDENTIFICATION=propusr
        OPERATOR-ROLE=proprole)
```

DELETE-PROP-OBJECT-DESCRIPTION

This command is used to delete an existing administration entity from the database.

It is provided for dynamic modification of the generation at runtime. Administration processes that are currently working with the specified administration entity are not affected by this command.

DELETE-PROP-OBJECT-DESCRIPTION
OBJECT-NAME = <structured-name 1..30>

OBJECT-NAME = <structured-name 1..30>

Name of the administration entity.

Return codes when the command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP0909	Internal error	Administration entity could not be deleted.
POP0102	Semantic error	No administration entity definition.

Table 5: Return codes of the DELETE-PROP-OBJECT-DESCRIPTION command

MODIFY-PROP-PARAMETERS

This command is used to:

- limit the resources of administration processes in order to minimize storage space and/or runtimes.

  Resource settings can be defined centrally for the PROP-XT subsystem. These quantities are operation-specific. If the command is reissued, only the specified parameters are modified.
- specify a file in order to store the database built up at runtime.
- specify an encryption password for additional backup of the passwords stored in the database.

```
MODIFY-PROP-PARAMETERS

 BUFFER-SIZE = *UNCHANGED / <integer 1000..100000>
,OBJECT-LIMIT = *UNCHANGED / <integer 1..100>
,PROCESS-LIMIT = *UNCHANGED / <integer 1..100>
,DATABASE-FILE = *UNCHANGED / *NO / <full-filename 1..54>
,ENCRYPTION-PASSWORD = *UNCHANGED / *PARAMETERS (...)

    *PARAMETERS (...)
       |   OLD-PASSWORD = *STD / <c-string 1..8> / <x-string 1..16>
       |  ,NEW-PASSWORD = *STD / <c-string 1..8> / <x-string 1..16>
```

BUFFER-SIZE = *UNCHANGED / <integer 1000..100000>

This operand specifies the maximum event buffer size (in bytes) per administration procedure.

When PROP-XT is started, the available storage space is set to 20000 bytes.

You must take into account that this buffer stores not only net messages, but also the mapping format of messages, plus 168 bytes of administration information per message event (OBJECT-NAME, REFERENCE-NAME, USER-ID, etc.).

OBJECT-LIMIT = *UNCHANGED / <integer 1..100>

This operand specifies the maximum number of administration entities supported per administration procedure. This is also the maximum number of administration entities supported by STD-ACTION.

When PROP-XT is started, the object limit is set to 100.

PROCESS-LIMIT = *UNCHANGED / <integer 1..100>
This operand specifies the number of administration procedures permitted in the system.

When PROP-XT is started, the process limit is set to 100.

DATABASE-FILE = *UNCHANGED / *NO / <full-filename 1..54>
This operand can be used to define or close a database file.

ℹ Since the connection passwords are stored in the database file (two-way encryption), it is safer not to work with a database file (see also page 16).

No database file is defined when PROP-XT is started.

*NO means that the database is maintained only in the memory and not in a database file.

If you specify a database file by means of DATABASE-FILE = <full-filename>, the current database is maintained in the memory and in the database file.
This means that CREATE-PROP-OBJECT-DESCRIPTION and DELETE-PROP- OBJECT-DESCRIPTION change both the database in the memory and the database file.
If you have worked with another database file up to then, this file is closed. The contents of the database will then be those at the time the switchover was made.

When a database file is opened, the current encryption password must be the same as the one contained in the database file. (Setting with ENCRYPTION-PASSWORD.)

Filing location for the database file:

- Entry via a logical console with a dynamic authorization name:
  - If <full-filename> contains a USER-ID, the file is created under this ID. The command originator has to be able to access this ID.
  - If <full-filename> does not contain a USER-ID, the file is created unter the ID of the console message.
- Entry via a logical console with a static authorization name and JOIN entry:
  - If <full-filename> contains a USER-ID, the file is created under this ID. The person making the entry has to be able to access this ID.
  - If <full-filename> does not contain a USER-ID, the file is created unter the ID corresponding to the JOIN entry.
- All other cases (i.e. entries via physical consoles or via logical consoles with a static authorization name and without a JOIN entry):
  - If <full-filename> contains a USER-ID, the file is created under this ID.
  - If <full-filename> does not contain a USER-ID, the file is created under TSOS.

ENCRYPTION-PASSWORD =
Encryption password for the stored connection passwords. Only one encryption password applies to all connection passwords. If you change the encryption password, all the stored passwords are re-encrypted.

If ENCRYPTION-PASSWORD and the database file are changed at the same time with the MODIFY-PROP-PARAMETERS command, PROP-XT first changes the file and then the encryption password. This means that the old database file will still contain the old encryption password.

ENCRYPTION-PASSWORD = *UNCHANGED
The encryption password remains unchanged.

ENCRYPTION-PASSWORD = *PARAMETERS (...)
The encryption password is changed.

OLD-PASSWORD = *STD / <c-string 1..8> / <x-string 1..16>
You have to specify the existing password to prove that you are authorized to make the change. *STD means that the default value is used by PROP-XT.

NEW-PASSWORD = *STD / <c-string 1..8> / <x-string 1..16>
The new encryption password is entered. *STD means that the default value is used by PROP-XT.

Return codes when command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP0707	Internal errors	Errors occurred in PROP-XT (DVS error or resource bottleneck). Possible causes: - The open database file cannot be closed. - The database file cannot be created, opened, read or written to. - The password cannot be encrypted.
POP0708		Errors occurred in PROP-XT (resource bottleneck, data inconsistency). The status of the database is undefined.
POP0229	Semantic errors	The passwords contained in the file were not encrypted with the specified password or the file is not a PROP-XT database file. The command was not executed.
POP0235		PROP-XT cannot work with the version specified in the database file or the file is not a PROP-XT database file. The command was not executed.
POP0705		The command was not executed. Changes to the database are only made if the valid password in entered.
POP022A	Warning	New database file was generated.

Table 6: Return codes of theMODIFY-PROP-PARAMETERS command

Beispiel

A maximum of five administration processes are to be able to run simultaneously.
The current database is to be maintained in the memory and in the file HUGO. The old encryption password was the default value used by PROP-XT. The encryption password to apply now is SCHLOSS17.

```
/MODIFY-PROP-PARAMETERS
  PROCESS-LIMIT=5,
  DATABASE-FILE=HUGO,
  ENCRYPTION-PASSWORD=*PARAMETERS
  OLD-PASSWORD=*STD, NEW-PASSWORD='SCHLOSS17'
```

REMOVE-PROP-STD-ACTION

This command is used to delete existing standard actions STD-ACTIONs from the database. From this point onwards, PROP-XT no longer sends the defined event.

You can only delete STD-ACTIONs which have already been defined, but which have not yet been deleted.

REMOVE-PROP-STD-ACTION
EVENT-NAME = <structured-name 1..30>

EVENT-NAME = <structured-name 1..30>
This operand specifies the ID of a STD-ACTION.

Return codes when the command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP0904	Internal error	STD-ACTION could not be deleted.
POP0903	Semantic error	STD-ACTION does not exist.

Table 7: Return codes of the REMOVE-PROP-STD-ACTION command

SEND-TO-PROP-PROCESS

This command can be used to request an administration process, to exchange an administration procedure (see also page 31) or to terminate itself.
After a termination request, the administration process is given with the next system command the corresponding return code 0A14 in the SYSPOP.MAINCODE variable.

SEND-TO-PROP-PROCESS
PROCESS-NAME = ***ALL** / list-poss(20): <structured-name 1..30>
,**TYPE** = ***STOP-REQUEST** / ***EXCHANGE-REQUEST** (...)
***EXCHANGE-REQUEST** (...)   ⎸ **NEW-PROCEDURE** = <c-string 1..256>

PROCESS-NAME = *ALL / list-poss(20): <structured-name 1..30>
This operand specifies which administration processes are addressed.

TYPE =
This operand defines the purpose for which the addressed administration processes are requested.

TYPE = *STOP-REQUEST
The administration process is requested to terminate itself.

TYPE = *EXCHANGE-REQUEST (...)
The administration process is requested to exchange an administration procedure.

NEW-PROCEDURE = <c-string 1..256>
Name of the file in which the new administration procedure is stored.
PROP-XT passes NEW-PROCEDURE on to the administration process without checking the existence of the file or an access protection. These checks are only carried out by SDF-P with INCLUDE-PROCEDURE. Since PROP-XT does not check the contents of the file, other information can be transferred in coordination with the administration procedure.

Runtime messages

Message code	Meaning
POP9903	None of the specified processes is known.
POP9908	Administration process ’(&01)’ could not be informed.
POP9909	Administration process ’(&01)’ could be informed.

Table 8: Messages of the SEND-TO-PROP-PROCESS command

Return codes when the command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP090B	Internal error	Command not executed fully.
POP090C	Semantic error	No administration process specified in command exists.
POP090D	Warning	Not all administration processes exist.

Table 9: Return codes of the SEND-TO-PROP-PROCESS command

Example

All administration processes should terminate themselves in a coordinated manner:

```
/SEND−TO−PROP−PROCESS
```

SHOW-PROP-DATABASE

This command outputs the descriptions of administration entities which exist in the database.

SHOW-PROP-DATABASE
OBJECT-NAME = ***ALL** / list-poss(20):<structured-name 1..30> ,**INFORMATION** = ***STD** / ***MAX**

OBJECT-NAME = *ALL / list-poss(20):<structured-name 1..30>
This operand is used to select the administration object(s) for which information is to be output.

INFORMATION = *STD / *MAX
If *STD is specified, only the name and type are output.
The specification *MAX outputs all existing information, with the exception of the passwords.

Runtime message

Message code	Meaning
POP990C	The specified object is not signed on to PROP-XT.

Table 10: Message of the SHOW-PROP-DATABASE

Return codes when the command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP0911	Semantic error	No administration entity specified in command exists.
POP0912	Warnings	Not all administration entities exist.
POP0913		No administration entity exists.

Table 11: Return codes of the SHOW-PROP-DATABASE command

Example

To output all information on the administration entities SYSTEMR2 and SYSTEMR3.

```
/SHOW-PROP-DATABASE OBJECT-NAME=(SYSTEMR2, SYSTEMR3),
/INFORMATION=*MAX
```

Output:

```
/OBJECT-NAME=SYSTEMR2
/   OBJECT=*OPERATING
/      PROCESSOR-NAME=RECHNER2
/      USER-IDENTIFICATION=HUGO2
/      OPERATOR-ROLE=PROPROLE
/   USER-ACCESS=*ALL-USERS
/OBJECT-NAME=SYSTEMR3
/   OBJECT=*OPERATING
/      PROCESSOR-NAME=*LOCAL
/      USER-IDENTIFICATION=*OWN
/   USER-ACCESS=MYID
```

SHOW-PROP-PARAMETERS

This command is used to query global parameters and data.

PROP-XT outputs information on any value that can be defined using the MODIFY-PROP-PARAMETERS command.

SHOW-PROP-PARAMETERS

This command has no operands.

Example

To output global information:

```
/SHOW-PROP-PARAMETERS
```

```
BUFFER-SIZE=50000
OBJECT-LIMIT=3
PROCESS-LIMIT=10
DATABASE-FILE=*NO
```

SHOW-PROP-PROCESS-STATUS

This command is used to provide information on administration processes signed on to PROP-XT.

SHOW-PROP-PROCESS-STATUS
PROCESS-NAME = *ALL / list-poss(20): <structured-name 1..30> ,**SEL**ECT = *STD / *OBJECTS / *EVENTS

PROCESS-NAME = *ALL / list-poss(20): <structured-name 1..30>

This operand specifies the administration entities on which information is to be output.

SELECT =

This operand specifies the information to be output for the administration processes defined in PROCESS-NAME.

SELECT = *STD

The TSN of all specified processes is output.

SELECT = *OBJECTS

A list of all the administration entities that are assigned to the specified processes is output.

SELECT = *EVENTS

The event definitions and the number of unretrieved events are output for the specified processes.

Runtime messages

Message code	Meaning
POP9903	None of specified processes known.
POP9904	Administration process ’(&01)’ not known.
POP9905	No administration entity exists for administration process ’(&01)’.
POP9906	No event exists for administration process ’(&01)’.

Table 12: Messages of the SHOW-PROP-PROCESS-STATUS command

Return codes when the command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP090B	Internal error	Command not executed fully.
POP090C	Semantic error	No administration process specified in command exists.
POP090D	Warning	Not all administration processes exist.

Table 13: Return codes of the SHOW-PROP-PROCESS-STATUS command

Example 1

To display all PROP-XT processes:

```
/SHOW-PROP-PROCESS-STATUS
```

```
PROCESS-NAME=PPM
  TSN=4711
PROCESS-NAME=TPM
  TSN=4713
PROCESS-NAME=XAF
  TSN=4715
ROCESS-NAME=EXIT123
  TSN=4789
PROCESS-NAME=HUGO
  TSN=4712
PROCESS-NAME=USER55
  TSN=4714
```

Example 2

To display all connections of the administration process PPM:

```
/SHOW-PROP-PROCESS-STATUS PROCESS-NAME=PPM,SELECT=*OBJECTS
```

```
PROCESS-NAME=PPM
  OBJECT-NAME=UCONZ
  OBJECT-NAME=UCONL
```

Example 3

To display the events of the PROP-XT process USER55:

```
/SHOW-PROP-PROCESS-STATUS PROCESS-NAME=USER55,SELECT=*EVENTS
```

```
PROCESS-NAME=USER55
  EVENT-NAME=LOGON
    EVENT-COUNT=0
  EVENT-NAME=LOGOFF
    EVENT-COUNT=0
  EVENT-NAME=EXC4233
    EVENT-COUNT=5
  EVENT-NAME=END-COM
    EVENT-COUNT=0
```

SHOW-PROP-STD-ACTION

This command is used to query the list of defined standard actions, which were defined using the ADD-PROP-STD-ACTION command (see page 38).

SHOW-PROP-STD-ACTION
EVENT-NAME = ***ALL** / list-poss(20): <structured-name 1..30>

EVENT-NAME = *ALL / list-poss(20): <structured-name 1..30>
This operand specifies whether all reactions (*ALL) are to be output, or only those specified in EVENT-NAME.

Runtime messages

Message code	Meaning
POP9902	STD-ACTION ’(&01)’ not known.

Table 14: Message of the SHOW-PROP-STD-ACTION command

Return codes when the command is terminated

MAINCODE	Error class	Meaning
CMD0001	Successful	Command executed successfully.
POP090B	Internal error	Command not executed fully.
POP0906	Semantic error	None of specified STD-ACTIONs exists.
POP0907	Warning	Not all specified STD-ACTIONs exist.

Table 15: Return codes of the SHOW-PROP-STD-ACTION command

Example

To output the reactions with the IDs MES-OMP0004 and MES-OMP0006:

```
/SHOW-PROP-STD-ACTION EVENT-NAME=(MES-OMP0004, MES-OMP0006)
```

```
EVENT-NAME=MES-OMP0004
  EVENT-DATA=SYSTEM-MSG
    MSG-ID=OMP0004
  ACTION=CMD
    OBJECT-NAME=STANDARD-ACTION-AO
    DATA=/CANCEL 4711

EVENT-NAME=MES-OMP0006
  EVENT-DATA=SYSTEM-MSG
    MSG-ID=OMP0006
  ACTION=CMD
    OBJECT-NAME=STANDARD-ACTION-AO
    DATA=/CANCEL 4711
```

4 Introduction to programming with PROP-XT

The user defines administration tasks in administration procedures using PROP-XT system commands. Reactions of varying complexity can be defined as responses to specific events.

4.1 Typical processes in an administration procedure

The formal structure of an administration procedure is outlined below:

```
SET-PROCEDURE-OPTIONS
DECLARE-VARIABLE SYSPOP(TYPE=STRUCTURE)
    BEGIN-PROP-PROCESS
        START-PROP-OBJECT-MONITORING ...
            START-PROP-EVENT-MONITORING ...
            WHILE (END=FALSE)
                WAIT-FOR-PROP-EVENTS ...
                Analysis and processing of events ...
                SEND-TO-PROP-ADMINISTRATOR ...
                SEND-TO-PROP-OBJECT ...
            END-WHILE
            STOP-PROP-EVENT-MONITORING ...
        STOP-PROP-OBJECT-MONITORING ...
    END-PROP-PROCESS
EXIT-PROCEDURE
```

DECLARE-VARIABLE

Following the definitions required for creating an SDF-P procedure (including the variables for the return information SYSPOP and possibly <RESULT-VARIABLE> with DECLARE-VARIABLE NAME=...), the administration process must first sign on to PROP-XT by means of

BEGIN-PROP-PROCESS

An administration entity from which messages are to be received or to which a message is to be sent, must be defined using the command

START-PROP-OBJECT-MONITORING

The events that are to initiate a reaction are defined with

START-PROP-EVENT-MONITORING

The events generated automatically by PROP-XT when processing message sequences are defined implicitly with the commands SEND-TO-PROP-OBJECT and SEND-TO-PROP-ADMINISTRATOR. The arrival of the defined events can be awaited with

WAIT-FOR-PROP-EVENTS

In response to an event, a message can be sent to an administrator with

SEND-TO-PROP-ADMINISTRATOR

or a command or the response to a question can be sent to an administration entity with

SEND-TO-PROP-OBJECT

If the response is to be the transmission of a message to an administration entity, this administration entity must be defined beforehand (with START-PROP-OBJECT-MONITORING). Event definitions defined with START-PROP-EVENT-MONITORING are deleted with

STOP-PROP-EVENT-MONITORING

along with all buffered events. The connection to the administration entity is cleared down with the command

STOP-PROP-OBJECT-MONITORING

Event definitions and events relating to this administration entity that have already been buffered, are deleted. The administration process is signed off from PROP-XT administration with the command

END-PROP-PROCESS

System commands (assignment and return data)

The transfer data to PROP-XT takes the form of system command operands.

Return data from the system commands is supplied in SDF-P variables: the SYSPOP variable is defined with the respective return codes after each system command is issued. If return information is also required in <RESULT-VARIABLE>, the name of the structured variable in which the required information is to be stored must be defined in the respective RESULT-VARIABLE command operand. The structure elements of SYSPOP and <RESULT-VARIABLE> are declared implicitly, i.e. when the value is assigned by PROP-XT. If RESULT-VARIABLE is not specified by the user, the return information is stored in SYSPOP.

The SYSPOP variable must be defined, as otherwise no return codes can be transferred to the administration procedure.

If variables in nested procedures (outside the procedure in which the command was issued) are to be accessed, the variables must be imported with IMPORT-VARIABLE in accordance with the SDF-P concept.

Activating the administration process

The administration process in which the administration procedure is to run must sign on to PROP-XT under its name using BEGIN-PROP-PROCESS. This command also carries out the authorization check, i.e. it checks whether the user ID under which the task is running has PROP-ADMINISTRATION privileges.

Defining events

The command START-PROP-EVENT-MONITORING is used to define events that are to be awaited with WAIT-FOR-PROP-EVENTS (see the section on “Definition of events”, page 21ff).

Incoming events are buffered by PROP-XT until they are picked up with the WAIT-FOR-PROP-EVENTS command. START-PROP-EVENT-MONITORING is used to define whether all incoming events or only those chronologically last are buffered and taken over by the administration procedure.

Waiting for events

With the WAIT-FOR-PROP-EVENTS command, the administration procedure is shifted to a wait state and is only reactivated when events occur (defined and job-related) or by a TIMEOUT. Detailed return information on type, origin, and possibly the message (command) itself is given in SYSPOP or <RESULT-VARIABLE>.

If an incoming event is described by multiple event definitions, the event is buffered a number of times and is passed to the administration procedure with the corresponding WAIT command.

Defining reactions

The return information supplied in WAIT-FOR-PROP-EVENTS determines the further course of the administration procedure. The logical process programmed with SDF-P is continued. It is possible to wait for further events to arrive. However, a message can also be sent to an administration entity (a command, the response to a question, or supplementary information) with SEND-TO-PROP-ADMINISTRATOR, or a message can be sent to the administrator (a message, a question, a command result, a request for supplementary information, or a command acknowledgment) with SEND-TO-PROP-ADMINISTRATOR.

Deactivating the administration process

With the END-PROP-PROCESS command, the administration process signs off from PROP-XT administration. The connection to the PROP-XT subsystem remains open until the task is ended, i.e. with the administration command SHOW-PROP-PROCESS-STATUS the administration process is no longer displayed, while with SHOW-SUBSYSTEM-STATUS the task TSN is displayed as still connected.

4.2 Processing messages

Using sample listings, the acceptance and transmission of messages of different types in administration procedures is explained below:

Accepting a message

The START-PROP-OBJECT-MONITORING command is used to define the administration entity from which the message is sent.

```
/ START-PROP-OBJECT-MONITORING -
/     OBJECT-NAME=SYSTEM
```

The message to which a response is required is signed on with START-PROP-EVENT-MONITORING

```
/ START-PROP-EVENT-MONITORING -
/      EVENT-NAME=UNKNOWN-STD-ACTION, -
/      SELECT-EVENT=FROM-OBJECT -
/      (EVENT-DATA=*SYSTEM-MSG -
/      (MSG-ID=POP9902))
```

and the arrival of this message is then awaited with WAIT-FOR-PROP-EVENTS:

```
/ WAIT-FOR-PROP-EVENTS -
/      EVENT-NAME=UNKNOWN-STD-ACTION, -
/      RESULT-VARIABLE=SYS-MSG
```

In the example, a system message (specified by its message code) is awaited.

It is only possible to await messages that are actually dispatched from UCON to this administration entity, i.e.

– the transmission of the relevant message must be unsolicited

– the message must be sent with a routing code for which the administration entity has receive authorization (appropriate operator role)

The incoming messages are stored in the structures of the return variables:

```
/ DECLARE-VARIABLE NAME=SYSPOP,TYPE=STRUCTURE
/ DECLARE-VARIABLE NAME=SYS-MSG,TYPE=STRUCTURE
```

The message code, message text, as well as the individual inserts can be accessed when analyzing the return variables. The entire message text (beginning with the message number) is stored after the UCON header as of position 19.
In the example, insert 1 and the time specification from the UCON header are processed further.

```
/ IF (SYSPOP.MAINCODE='0000')
/ SET-VARIABLE STD-ACTION-NAME=SYS-MSG.I1
/ SET-VARIABLE TIME=SUBSTRING(STRING=SYS-MSG.INFORMATION, -
/                             START=11,LENGTH=6)
/ END-IF
```

where

```
/ DECLARE-VARIABLE NAME=STD-ACTIONS-NAME,TYPE=STRING
/ DECLARE-VARIABLE NAME=TIME,TYPE=STRING
```

If a message of type FREE-STRING is accepted, the incoming message is compared with the character string defined by START-PROP-EVENT-MONITORING, whereby the incoming message is prefixed by a UCON header so that the text begins at position 18:

```
/ START-PROP-EVENT-MONITORING -
/      EVENT-NAME=NOTE, -
/      SELECT-EVENT=*FROM-OBJECT -
/      (EVENT-DATA=*FREE-STRING -
/      (POSITION=18,STRING='NOTE'))
```

PROP-XT offers a special handling facility for DADM messages:
Normally, a DADM message is split into numerous physical message segments. PROP-XT combines these message segments together so that a single WAIT command can be used to receive one logical DADM message.
If a DADM message comprising a number of message segments is to be accepted, it is necessary to specify the position in the combined message, i.e. the length of the preceding message segments must be added to the position in the message segment.

Sending a message

A message is sent to the administrator with the SEND-TO-PROP-ADMINISTRATOR command.

If a system message is to be sent, it must be ensured that the message associated with the message number is contained in a message file.

```
/ SEND-TO-PROP-ADMINISTRATOR -
/      ADMINISTRATOR-NAME=*BY-OPERATOR-ROLE(ROLE-ID='R'), -
/      DATA=*SYSTEM-MSG(MSG-ID=POP0A02)
```

If the data format FREE-STRING is used, the message text is not prefixed by a header.

```
/ SEND-TO-PROP-ADMINISTRATOR -
/      ADMINISTRATOR-NAME=*BY-OPERATOR-ROLE(ROLE-ID='R'), -
/      DATA='MESSAGE TO THE ADMINISTRATOR'
```

Accepting and answering a question

The START-PROP-OBJECT-MONITORING command is used to define the administration entity from which the question is sent.

```
/ START-PROP-OBJECT-MONITORING -
/     OBJECT-NAME=SYSTEM
```

The question (message) to which a reaction is required, signs on with START-PROP-EVENT-MONITORING.

```
/ START-PROP-EVENT-MONITORING -
/      EVENT-NAME=UNKNOWN-STD-ACTION, -
/      SELECT-EVENT=*FROM-OBJECT -
/      (EVENT-DATA=*SYSTEM-MSG -
/      (MSG-ID=IDA0N52))
```

The arrival of this question is then awaited with WAIT-FOR-PROP-EVENTS.

```
/ WAIT-FOR-PROP-EVENTS -
/      EVENT-NAME=UNKNOWN-STD-ACTION, -
/      RESULT-VARIABLE=SYS-QUESTION
```

In the example, a question of type system message (identified by its message code) is awaited. Similarly, a question (message) of type FREE-STRING could be awaited. The same rules apply to the analysis of message texts as to the acceptance of messages.

The incoming questions (messages) are stored in the structures of the return variables:

```
/ DECLARE-VARIABLE NAME=SYSPOP,TYPE=STRUCTURE
/ DECLARE-VARIABLE NAME=SYS-QUESTION,TYPE=STRUCTURE
```

In response to the incoming message, the text ‘BAD’ is sent to the administration entity with the SEND-TO-PROP-OBJECT command.

```
/ IF ((SYSPOP.MAINCODE='0000') AND -
/     (SYS-QUESTION.INFORMATION-TYP='QUESTION'))
/ SEND-TO-PROP-OBJECT  -
/      OBJECT-NAME=SYSTEM, -
/      TYPE=REPLY, -
/      DATA='BAD', -
/      REFERENCE-NAME=&(SYS-QUESTION.REFERENCE-NAME)
/ END-IF
```

In the response to a question (REPLY), the name transferred either when the message was received or in the return variable (SYS-QUESTION.REFERENCE-NAME in the example) must be specified as the REFERENCE-NAME in the SEND-TO-PROP-OBJECT command.

Sending a question

As with a message, a question is sent to the administrator using the SEND-TO-PROP-ADMINISTRATOR command.

```
/ SEND-TO-PROP-ADMINISTRATOR -
/      ADMINISTRATOR-NAME=*BY-OPERATOR-ROLE(ROLE-ID='R'), -
/      TYPE=*QUESTION, -
/      DATA=*SYSTEM-MSG(MSG-ID=POP0A02)
```

As regards the data formats (transmission of a system message specifying the message number or FREE-STRING), the same rules apply as for messages.

The REFERENCE-NAME operand can be specified in the SEND-TO-PROP-ADMINISTRATOR command. The name specified by REFERENCE-NAME is used as the job ID for the administrator when a question is transmitted. At the same time, a job-related event is generated with this name. With REFERENCE-NAME as the EVENT-NAME, the response can be awaited with the WAIT-FOR-PROP-EVENTS command.

If questions are issued by several administration procedures simultaneously, the uniqueness of the REFERENCE-NAME can no longer be ensured by the administration procedure. If no REFERENCE-NAME is specified (*NO), PROP-XT ensures that a question is assigned a unique job ID, which is set up and managed by PROP-XT. The administration procedure is notified of a REFERENCE-NAME formed in this way by means of the return variable.

Accepting a command

An administration procedure can accept user-defined commands. These must be defined beforehand with SDF-A. The definition must be contained in a group syntax file, which is assigned both to the respective user and to the ID under which the administration procedure is to run.

```
ADD-COMMAND ASSIGN-STD-PRINTER,IMPLEMENTOR=TPR(ENTRY=DUMMY,CALL=NEW)
ADD-OPERAND NAME=PRINTER...
```

The user-defined command formulated in this way can be entered by the administrator and, if the authorization check is successful (the administrator is authorized to enter the command), can be accepted by the administration procedure, if it is defined with START-PROP-EVENT-MONITORING:

```
/ START-PROP-EVENT-MONITORING -
/      EVENT-NAME=ASP, -
/      SELECT-EVENT=*FROM-ADMINISTRATOR -
/      (CMD-COMPLETITION=*YES, -
/      CMD-NAME=ASSIGN-STD-PRINTER)
```

CMD-COMPLETITION=*YES defines that the administration procedure must send a command acknowledgment in order to conclude the command. If no command acknowledgment is sent (*NO), no results can be sent either.

The command entered by the administrator can now be awaited with WAIT-FOR-PROP-EVENTS (ASSIGN-STD-PRINTER in the example):

```
/ WAIT-FOR-PROP-EVENTS -
/      EVENT-NAME=ASP, -
/      TIME-LIMIT=NO
```

After the command has been executed successfully, the command acknowledgment must be sent to the administrator. Here, the name specified in SYSPOP.REFERENCE-NAME must be in the REFERENCE-NAME operand.

```
/ IF (SYSPOP.MAINCODE='0000')
/ SET-VAR REFNAM=SYSPOP.REFERENCE-NAME
/ SEND-TO-PROP-ADMINISTRATOR -
/      TYPE=*CMD-TERMINATION(), -
/      DATA=C'CMD0001', -
/      REFERENCE-NAME=&REFNAM
```

In the command acknowledgment (TYPE=*CMD-TERMINATION) that must be sent to the administrator, UCON expects the DATA operand to contain only a message number (not a message!), which in this case must be transferred as a character string, not as a message. UCON sets up the command acknowledgment itself, whereby the transferred message number is used as an insert. The sender must ensure that the message associated with the message number is contained in a message file.

If an administration procedure signs on for processing a command for which another procedure has already signed on, the command is transferred as of this point to the last administration procedure to sign on. From the point of view of UCON, the administration procedure acts as a command server for operator special commands: if four command servers have already signed on for the command, any attempt of another command server to sign on will be rejected.

Sending a command

The administration entity to which the command is to be sent is defined with the START-PROP-OBJECT-MONITORING command.

```
/ START-PROP-OBJECT-MONITORING -
/     OBJECT-NAME=SYSTEM
```

The command (the administration command SHOW-PROP-PARAMETERS in this example) is sent to the administration entity with SEND-TO-PROP-OBJECT.

```
/ SEND-TO-PROP-OBJECT -
/      OBJECT-NAME=SYSTEM, -
/      TYPE=CMD(IMPLICIT-EVENTS=*ALL), -
/      DATA='SHOW-PROP-PARAMETERS', -
/      REFERENCE-NAME=PROP-PARAM
```

If IMPLICIT-EVENTS=*ALL is specified, implicit job-related events are defined and generated, and are transferred to the administration procedure with the WAIT-FOR-PROP-EVENTS command. If IMPLICIT-EVENTS=*CMD-TERMINATION, only the command acknowledgment can be specified.
Job-related events are always identified by the job ID (REFERENCE-NAME). If you want to receive command results (IMPLICIT-EVENTS=*ALL or *CMD-TERMINATION), then REFERENCE-NAME must be specified.

The return information (including command results) supplied with the WAIT-FOR-PROP-EVENTS command can be queried in a loop in the structures of the return variables, and appropriate reactions can be defined.

```
/ DECLARE-VARIABLE NAME=SYSPOP,TYPE=STRUCTURE
/ DECLARE-VARIABLE NAME=END,TYPE=BOOLEAN,INITIAL-VALUE=FALSE
/ DECLARE-VARIABLE NAME=OBJECT-LIMIT,TYPE=STRING

/ WHILE (END=FALSE)
/    WAIT-FOR-PROP-EVENTS -
/         EVENT-NAME=PROP-PARAM
/    IF (SYSPOP.MAINCODE='0000')
/       IF (SYSPOP.INFORMATION-TYP='CMD-RESULT')
/          IF (SUBSTRING(STRING=SYSPOP.INFORMATION,START=18,LENGTH=3) -
/             ='OBJ')
/             SET-VARIABLE OBJECT-LIMIT=SUBSTRING(STRING=SYSPOP.INFORMATION, -
/                                                 START=31,LENGTH=6)
/             ...
/          END-IF
/       ELSE-IF (SYSPOP.INFORMATION-TYP='ADDITIONAL-INFORMATION-REQUEST')
/          REMARK KEINE AKTION

/       ELSE
/          REMARK (SYSPOP.INFORMATION-TYP='CMD-TERMINATION')
/          SET-VARIABLE END=TRUE
/       END-IF
/    END-IF
/ END-WHILE
```

Some commands send their “results” not as messages as part of command processing, but as unsolicited messages (e.g. the LOGON command sends the message JMS0154). Such results must be reported explicitly with START-PROP-EVENT-MONITORING, before the command is issued so that they are not lost. They are awaited with WAIT-FOR-PROP-EVENTS.

PROP-XT does not guarantee that an unsolicited “command result” message received with WAIT-FOR-PROP-EVENTS actually belongs to the command issued by the administration procedure. It could also belong to another command issued by another administrator at around the same time. The administration procedure itself is responsible for performing appropriate checks.

4.3 General programming information

WAIT-FOR-PROP-EVENTS wait point

Events that are to be awaited with WAIT-FOR-PROP-EVENTS sign on to PROP-XT with the START-PROP-EVENT-MONITORING command. Incoming events are buffered by PROP-XT. The incoming events are not transferred to the administration procedure in accordance with their event ID until WAIT-FOR-PROP-EVENTS is issued.

The administration procedure can define how incoming events are to be buffered:

- All incoming events are buffered and are passed to the administration procedure.
- Only the last event to occur is passed to the administration procedure. This means that a newer event overwrites one already buffered if the same event description applies.

Buffered events that are not picked up can lead to a buffer overflow. In this case, incoming events are lost.

Implicitly defined events generated during command processing (“command-related events”) must also be picked up with WAIT-FOR-PROP-EVENTS.

Events can be deleted using the following commands:

- STOP-PROP-EVENT-MONITORING deletes event definitions defined with START-PROP-EVENT-MONITORING, together with all buffered events. This means that events corresponding to the defined event definition, and all events that have already arrived for this event definition, are not dispatched.
- STOP-PROP-OPBJECT-MONITORING deletes an administration entity together with all buffered events and the event definitions relating to this event.

All commands that shift the task to a wait state (e.g. ON, WAIT, PAUSE) are problematic, because the administration process thereby takes control from PROP-XT; the only permitted wait point in PROP-XT is visible on the programming interface with the WAIT-FOR-PROP-EVENTS command.
However, an administration procedure can survive very well without these commands:

- instead of ON and WAIT:
  Define job variable events with START-PROP-EVENT-MONITORING, pick up a job variable event with WAIT-FOR-PROP-EVENTS.
- instead of PAUSE:
  Send questions to the operator using SEND-TO-PROP-ADMINISTRATOR, and pick up replies with WAIT-FOR-PROP-EVENTS.

Performance of PROP-XT

When defining an administration entity, it must be remembered that a connection to UCON is established (and occupied) for every definition, although the total number of connections to UCON is limited.
If operator roles are assigned, all messages run via the connection sent with the routing codes contained in the operator roles.
For example, if five connections are occupied, all messages are transferred to PROP-XT five times and subsequently discarded by PROP-XT if they are not passed to the administration procedure. If message traffic is busy, response time is thus severely affected by unnecessary definitions. As a counter-measure, administration entities should only be defined sparingly, and operator roles should be assigned by the system administrator in accordance with the specific tasks.

5 Programming interface

This chapter provides details on the programming interface. In addition to PROP-XT system commands, it describes interface operation including error handling and test systems. A sample administration procedure also provides information on using the interface.
The chapter “User-defined commands“ on page 139ff describes how you can define your own commands.

5.1 Operation

Administration procedures call PROP-XT functions using the PROP-XT system commands, which are analyzed by SDF and processed by PROP-XT routines. PROP-XT transfer data takes the form of command operands, which are described in detail in the section “System commands” on page 81ff.

Return information from PROP-XT commands is supplied in SDF-P variables.
There are two groups of return information:

- The SYSPOP variable (SYSPOP.MAINCODE, SYSPOP.SUBCODE1, SYSPOP.SUBCODE2) is supplied with the respective error code each time a system command is issued.
- With the commands CHECK-PROP-EVENT-OCCURRENCE, SEND-TO-PROP-ADMINISTRATOR, and WAIT-FOR-PROP-EVENTS, the name of the structured variable in which the required return information (which may be extensive) is to be stored, can be defined using the RESULT-VARIABLE operand. If RESULT-VARIABLE is not specified, the result information is stored in SYSPOP.

All command-specific return codes and information are described in detail for the individual commands. Error situations that may occur with all PROP-XT system commands are explained in the section “Error handling” on page 79.

The return variables SYSPOP and <RESULT-VARIABLE> must be declared as structured variables with DECLARE-VARIABLE in the administration procedure. The structure elements of SYSPOP and <RESULT-VARIABLE> are declared implicitly, i.e. when the value is assigned by PROP-XT.

If return codes for a previous command were already supplied in <RESULT-VARIABLE>, the values irrelevant to the current command remain unchanged.

The return variables are defined as soon as an error occurs. In this case, an error message is also output to SYSOUT, the contents of which are determined by the message code transferred in SYSPOP.MAINCODE.

PROP-XT allows you to use SDF.A to define your own commands, which can be entered by the administrator. These user-defined commands are available in addition to the commands described in the section “System commands” on page 81ff. PROP-XT transfers these commands to the administration procedures for processing. The chapter “User-defined commands“ (page 139) describes how to handle these commands.

5.2 Error handling

In the event of errors at execution time, the reaction of PROP-XT is the usual response for SDF-P, see the manual “SDF-P”:
If errors occur during operation, error branches can be executed in the administration procedure by means of the SPIN-OFF mechanism or the SDF-P commands IF-BLOCK-ERROR and IF-CMD-ERROR. For the user, this means errors resulting from PROP-XT system commands can be handled with the IF-BLOCK-ERROR or IF-CMD-ERROR construct, see the manual “SDF-P”.

With errors other than SDF syntax errors, the SDF-P variable SYSPOP is assigned error codes: SYSPOP.MAINCODE (TYPE=(STRING)), SYSPOP.SUBCODE1 (TYPE=(INTEGER)), and SYSPOP.SUBCODE2 (TYPE=(INTEGER)).
The hexadecimal code specified in MAINCODE corresponds to the error number, see the chapter “Messages” on page 157ff.
The MAINCODE is 0000 is the case of successful execution.

The error codes correspond to the general return codes of SDF commands, whereby the following error classes are significant for PROP-XT:

SUBCODE1	**SUBCODE2**	**Error class**
0	0	Successful
1	0	Syntax error
32	0	Internal error
64	0	Semantic error
0	2	Warning

Table 16: General meanings of error classes

The following error situations may occur with all system commands:

MAINCODE	**Error class**	**Meaning**
0A0F	Syntax errors	Total length of inserts too great. The combined length of all the inserts in a system message may not exceed 180 characters.
0A10		Syntax error in command.
0A0E	Internal errors	No such PROP-XT system command in syntax file.
0A03		Resource bottleneck on system.

Table 17: Return codes applicable to all system commands (Section 1 of 2)

MAINCODE	Error class	Meaning
0A0C	Semantic errors	The administration procedure must start with BEGIN-PROP-PROCESS.
0A0D		Specified command is not a PROP-XT system command.
0A11		A second BEGIN-PROP-PROCESS command is only permitted in an administration task after an END-PROP-PROCESS command.

Table 17: Return codes applicable to all system commands (Section 2 of 2)

Administration procedures can give rise to error situations that do not relate directly to a command. Such error situations may prevent the administration procedure from continuing properly.

The administration procedure can check the status of PROP-XT using the SDF-P variable SYSPOP.MAINCODE. In error situations that prevent PROP-XT from continuing operation, the SYSPOP.MAINCODE variable is assigned as a last resort in each procedure.

The following error situations, which are not directly related to the last command, are possible:

MAINCODE	Error class	Meaning
0A1D	Internal error	Error in connection to one or more administration entity. This error is reported as soon as an attempt is made to send a message to this administration entity or if an event is awaited from this administration entity.
0A13	Semantic error	Arriving events can no longer be buffered for the administration process. The memory space required exceeds the value permitted by the administration command MODIFY-PROP-PARAMETERS. The error is reported with the next system command.
0A14	Warning	Termination requested. Via the administration command SEND-TO-PROP-PROCESS the administration process was requested to terminate. The warning is output with the next system command.

Table 18: Return codes not related directly to the last system command

5.3 System commands

The notational conventions used in the following command descriptions are explained in the section “Notational conventions” on page 5ff.

BEGIN-PROP-PROCESS

An administration process signs on to PROP-XT with this command. It must be the first PROP system command to be issued; before this command, all other PROP-XT system commands are rejected.

This command also carries out the authorization check, see section “Authorization checks” on page 34ff.

BEGIN-PROP-PROCESS
PROCESS-NAME = <structured-name 1..30>
,**OPERATIONAL-MODE** = ***OPERATION** / ***SIMULATION**

PROCESS-NAME = <structured-name 1..30>
This operand identifies the administration process. If an administration process of the specified name already exists, the request to sign on to PROP-XT is rejected.

OPERATIONAL-MODE =
This operand specifies whether the administration process is to be administered in reality, or whether it is simply to be tested.

OPERATIONAL-MODE = ***OPERATION**
The administration process is administered in reality.

OPERATIONAL-MODE = ***SIMULATION**
The administration process is not administered in reality, rather is just tested. The procedure for testing administration processes is described in the section “Test system” on page 135ff.

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A02	Internal errors	Memory bottleneck. The message buffer (MTBUFF) for communicating with the main task cannot be created.
0A03		Resource bottleneck on system: Bourse error in communication between administration task and main task. System error in querying the user ID of the administration process. System error in querying the TSN of the administration process.
0A16		Error in accessing administration entity management data.
0A17		Error in accessing event management data.
0A18		Error in accessing administration process management data.
0A00	Semantic errors	The user ID which was used to try to sign on to PROP-XT does not have the PROP-ADMINISTRATION privilege.
0A01		Administration process with this name has already signed on.
0108		No more administration processes/entities allowed.

Table 19: Error return codes of the BEGIN-PROP-PROCESS command

Example

Signing on to PROP-XT:

```
/BEGIN-PROP-PROCESS PROCESS-NAME=PROPPROZ1
```

CHECK-PROP-EVENT-OCCURRENCE

This command allows you to check whether events have occurred. When the command is executed, PROP-XT checks all or particular (selected) events.

PROP-XT distinguishes between the definition of two types of event:

- explicitly defined events declared with the START-PROP-EVENT-MONITORING command
- implicitly defined events (job events) identified by the job ID

See also section “Events” on page 21f.

CHECK-PROP-EVENT-OCCURRENCE
EVENT-NAME = *ALL / list-poss(20): <structured-name 1..30>
,RESULT-VARIABLE = *STD-VARIABLE / <structured-name 1..20>

EVENT-NAME = *ALL / list-poss(20): <structured-name 1..30>

This operand is used to specify an event ID. Explicitly defined events can be selected either using the explicitly defined identification (see START-PROP-EVENT-MONITORING, page 101 ff) or using the group identification. Job events, i.e. implicitly defined events (command result, reply to a question) are selected by means of their job ID.

RESULT-VARIABLE = *STD-VARIABLE / <structured-name 1..20>

Name of the return variable. This operand defines whether the events are transferred in the SYSPOP variable (*STD-VARIABLE) or in a variable specified here (<structured-name 1..20>).

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A17	Internal errors	Error in accessing event management data.
020F		Subsystem JV not available.
0A08	Semantic errors	None of the specified events defined for administration process.
0225		Access to job variable not possible.
0A1F	Warnings	Not all specified events are defined. Message POP9A04: “Event ’(&01)’ not known” is output for each such event.
0B13		Time specified for time event already elapsed.
0B18		No events occurred for selection criterion: No more events have occurred for the specified events since those that have already been fetched/transferred.

Table 20: Error return codes of the CHECK-PROP-EVENT-OCCURRENCE command

The structured variable <RESULT-VARIABLE> is defined as follows:

- <RESULT-VARIABLE>.EVENT-COUNT: Number of selected events that have occurred (TYPE = INTEGER).

  ℹ A job variable event is counted as having occurred if the job variable has the desired value at the time of the CHECK. The number of times the desired value has been modified since START-PROP-EVENT-MONITORING is not counted. The decisive factor is the current state (=value) of the job variable. Job variable events thus behave differently to all other events.

- <RESULT-VARIABLE>.EVENT-NAME: Event identification of the last event selected (TYPE = (STRING)).
- <RESULT-VARIABLE>.GROUP-NAME: Group identification of the last event selected (TYPE = (STRING)).

- <RESULT-VARIABLE>.EVENT-TYP: Type of the last event selected (TYPE = (STRING)).

  The following values are possible:

TIMER	time event
JV	job variable event
FROM-ADMINISTRATOR	message from administrator
FROM-AO	message from administration entity
ERROR	error

Table 21: Values of the EVENT-TYP structure

- <RESULT-VARIABLE>.INFORMATION-TYP: Message type of the last event selected (TYPE = (STRING)). This variable is not relevant to time and job variable events (nor errors), and neither is it defined in such cases.

  The following values are possible:

CMD	command
REPLY	reply
ADD-INFO	additional information
MSG	message
QUESTION	question
CMD-RESULT	command result
ADDITIONAL-INFORMATION- REQUEST	additional information request
CMD-TERMINATION	command termination acknowledgment

Table 22: Values of the INFORMATION-TYP structure

- <RESULT-VARIABLE>.PARTNER-NAME: Name of the partner that initiated the last event selected (TYPE = (STRING)).

  Specifies the name of the administration entity from which the event originated. If this is a message from an administrator managed by UCON, the variable contains the value 'ADMINISTRATOR'.

Example

Check whether DUMP messages have arrived. The subsequent analysis must be performed with SDF-P resources:

```
/CHECK-PROP-EVENT-OCCURRENCE EVENT-NAME=DUMP
```

END-PROP-PROCESS

This command is the last PROP-XT system command, and it occurs at most once in an administration procedure. It is initiated implicitly when the task terminates, if it is not available at the procedure.

It signs off the administration process from PROP-XT administration, clears down the connection to the administration entities, and terminates event buffering.

The task is not separated from the PROP-XT subsystem until the task ends.

END-PROP-PROCESS

This command has no operands.

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A03	Internal errors	Resource bottleneck on system: Bourse error in communication between administration task and main task. Error in releasing AP name for NAME-MANAGER. Error in releasing communication buffer.
0A16		Error in accessing administration entity management data.
0A17		Error in accessing event management data.
0A18		Error in accessing administration process management data.

Table 23: Error return codes of the END-PROP-PROCESS command

Example

Signing off from PROP-XT:

```
/END-PROP-PROCESS
```

SEND-TO-PROP-ADMINISTRATOR

Information is sent to the administrator in the form of a message:

- A directed, solicited message is sent to the administrator, who initiates the job identified by the specified ID.
- A non-directed, unsolicited message is sent to the administrators defined in the command.

PROP-XT supports UCON message types for administration entities of the OPERATING type.

(Section 1 of 2)

```
SEND-TO-PROP-ADMINISTRATOR
---------------------------------------------------------------------------------------
 ADMINISTRATOR-NAME = *BY-OPERATOR-ROLE (...)

    *BY-OPERATOR-ROLE(...)
       |   ROLE-ID = *FROM-MSG-FILE / <c-string 1..1>

,TYPE = *MSG / *QUESTION / *CMD-RESULT / *ADDITIONAL-INFORMATION-REQUEST /
    *CMD-TERMINATION(...)

    *CMD-TERMINATION(...)
       |  RETCO-SUBCODE2 = *SUCCESSFUL / *WARNING
       |  RETCO-SUBCODE1 = *SUCCESSFUL / *SYNTAX-ERROR / *INTERNAL-ERROR /
       |     *SEMANTICAL-ERROR
```

(Section 2 of 2)

```
,DATA = *SYSTEM-MSG (...) / <c-string 1..1800> / <x-string 1..3600>

   *SYSTEM-MSG(...)
   |    MSG-ID = <name 7..7>
   |   ,INSERT-0 = *NO / <c-string 1..180>
   |   ,INSERT-1 = *NO / <c-string 1..180>
   |   ,INSERT-2 = *NO / <c-string 1..180>
   |   ,INSERT-3 = *NO / <c-string 1..180>
   |   ,INSERT-4 = *NO / <c-string 1..180>
   |   ,INSERT-5 = *NO / <c-string 1..180>
   |   ,INSERT-6 = *NO / <c-string 1..180>
   |   ,INSERT-7 = *NO / <c-string 1..180>
   |   ,INSERT-8 = *NO / <c-string 1..180>
   |   ,INSERT-9 = *NO / <c-string 1..180>
   |   ,INSERT-10 = *NO / <c-string 1..180>
   |   ,INSERT-11 = *NO / <c-string 1..180>
   |   ,INSERT-12 = *NO / <c-string 1..180>
   |   ,INSERT-13 = *NO / <c-string 1..180>
   |   ,INSERT-14 = *NO / <c-string 1..180>

,REFERENCE-NAME = *NO / <structured-name 3..30>

,RESULT-VARIABLE = *STD-VARIABLE / <structured-name 1..20>
```

ADMINISTRATOR-NAME = *BY-OPERATOR-ROLE (...)

ROLE-ID = *FROM-MSG-FILE / <c-string 1..1>
Defines the routing code with which the unsolicited message is to be sent. *FROM-MSG-FILE means that the routing code is taken for selection from the message file.

TYPE = *MSG / *QUESTION / *CMD-RESULT / *ADDITIONAL-INFORMATION-REQUEST / *CMD-TERMINATION
The message type is specified as a keyword.

TYPE = *CMD-TERMINATION (...)
Before the command acknowledgment is sent, all job-related events ordered during command processing must be picked up.

RETCO-SUBCODE2 = *SUCCESSFUL / *WARNING
If the administration procedure operates as a command server, it defines whether it will send a command acknowledgment when signing on the command as an event. In accordance with BS2000 conventions, a command server must terminate command processing with particular, defined return codes.
RETCO-SUBCODE1 and 2 appear in the CMD-TERMINATION on the console as Insert1 and Insert2 respectively.

If an invalid type is specified (e.g. *ADDITIONAL-INFORMATION-REQUEST, although CMD signed on with CMD-INFO-REQUEST = *NO), UCON does not deliver the message. PROP-XT nevertheless supplies the (synchronous) return code 0 (SYSPOP.MAINCODE = 0).

RETCO-SUBCODE1 = *SUCCESSFUL / *SYNTAX-ERROR / *INTERNAL-ERROR / *SEMANTICAL-ERROR
See above description of RETCO-SUBCODE2.

DATA = *SYSTEM-MSG (...)/ <c-string 1..1800> / <x-string 1..3600>
Message in the form of a character string or in the form of message codes and inserts.

In the command acknowledgment (TYPE=*CMD-TERMINATION) that must be sent to the administrator, UCON expects only a message number (not a message!), which in this case must be transferred as a character string, not as a message. UCON sets up the command acknowledgment itself, whereby the transferred message number is used as an insert.

The sender must ensure that the message associated with the message number is contained in a message file, so that the command originator can obtain more detailed information on this message when the command terminates, using HELP-MESSAGE-INFORMATION. With free data formats, the administration process need not prefix the message with the necessary UCON headers; PROP-XT looks after this.

MSG-ID = <name 7..7>
Message code.

INSERT-i = *NO / <c-string 1..180>
INSERT-i identifies the message insert with the logical number i (where 0 ≤ i ≤ 14). The value *NO means that the value of insert i is irrelevant.

i The combined length of all inserts must not exceed 180 characters.

REFERENCE-NAME = *NO / <structured-name 3..30>
This operand identifies associated messages (message sequences with question / reply or command / result / additional information request / additional information / termination acknowledgment).

If the administration procedure assigns a name, the latter must not begin with the reserved character $ or #.

If a question (*QUESTION) is sent, the name specified here is used as the job ID for the administrator. UCON requires that the first three characters of this name be unique, because to respond the administrator identifies the question by its origin (= PROP-XT) + three-position job ID. A job-related event is also generated with this name, i.e. it must also be ensured that the name is unique as an EVENT-NAME. With REFERENCE-NAME as the EVENT-NAME, the reply can be awaited with the WAIT-FOR-PROP-EVENTS command.

If questions are issued by several administration procedures simultaneously, the uniqueness of the REFERENCE-NAME can no longer be ensured by the administration procedure. If no REFERENCE-NAME is specified (*NO), PROP-XT ensures that a question is assigned a unique job ID, which is set up and managed by PROP-XT. The administration procedure is notified of a REFERENCE-NAME formed in this way by means of the <RESULT-VARIABLE>.

With command messages (*CMD-RESULT, *CMD-TERMINATION, *ADDITIONAL-INFORMATION-REQUEST), the name transferred in <RESULT-VARIABLE>.REFERENCE-NAME on receipt of the command must be used as the REFERENCE-NAME. If an additional information request is issued (*ADDITIONAL-INFORMATION-REQUEST), a job-related event is generated with this name. The additional information can be awaited with WAIT-FOR-PROP-EVENTS using the name specified here. PROP-XT ensures that the names are unique and that no conflicts occur with the names assigned by the administration procedure ($ as first character). The implicitly generated job-related event has the name transferred in <RESULT-VARIABLE>.REFERENCE-NAME and belongs to the $$TRANSACTION group.

RESULT-VARIABLE = *STD-VARIABLE / <structured-name 1..20>
This operand defines whether REFERENCE-NAME is transferred in the SYSPOP variable (specification of *STD-VARIABLE) or in a variable defined here (specification of <structured-name 1..20>) .

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A03	Internal errors	Resource bottleneck on system: Bourse error in communication between administration task and main task.
0A1C		Port signals error when sending over administration connection.
0A17		Error in accessing event management data.
0213		Specified system message could not be generated.
0B0A	Semantic errors	Unknown REFERENCE-NAME
0B0B		Although job events are expected, no REFERENCE-NAME was given.
0B08		Specified REFERENCE-NAME already exists. Command ignored.
0B09		Resource bottleneck with internal PROP-XT reference name.
0B12		Message type currently not allowed.
010C		Syntax error in name.

Table 24: Error return codes of the SEND-TO-PROP-ADMINISTRATOR command

The structured variable <RESULT-VARIABLE> is defined as followed:

<RESULT-VARIABLE>.REFERENCE-NAME: ID of the job to which something was just sent (TYPE = (STRING)).

Example

Sending the unsolicited message “Warning: SHUTDOWN in 10 minutes!” to the console:

```
/SEND-TO-PROP-ADMINISTRATOR DATA='Warning: SHUTDOWN in 10 minutes!'
```

SEND-TO-PROP-OBJECT

Information is sent to the specified administration entity in the form of a message. UCON message types are supported..

```
SEND-TO-PROP-OBJECT

 OBJECT-NAME = <structured-name 1..30>

,TYPE = *CMD (...)/ *REPLY / *ADDITIONAL-INFORMATION

    *CMD(...)
      |    IMPLICIT-EVENTS = *ALL / *NO / *CMD-TERMINATION

,DATA = *NONE / <x-string 0..1000> / <c-string 0..500>

,REFERENCE-NAME = *NO / <structured-name 3..30>
```

OBJECT-NAME = <structured-name 1..30>
Name of the administration entity that is to receive the message.

TYPE =
UCON message type sent to the administration entity.

TYPE = *CMD (...)
A command is sent to the administration entity.

This implicitly defines events that are also delivered to the administration procedure (job events). As with the occurrence of explicitly defined events, the occurrence of job-related events can be defined with the CHECK-PROP-EVENT-OCCURRENCE command. Job events are likewise transferred with WAIT-FOR-PROP-EVENTS. They are always identified by the job ID (REFERENCE-NAME).

> **IMPLICIT-EVENTS =**
> Defines which command results the administration procedure wants to receive.
>
> **IMPLICIT-EVENTS = *ALL**
> All command-related messages are desired.
>
> **IMPLICIT-EVENTS = *NO**
> No command-related messages are desired.
>
> **IMPLICIT-EVENTS = *CMD-TERMINATION**
> Only the command acknowledgment is desired.

TYPE = *REPLY
The reply to a question is sent to the administration entity. PROP-XT adds the TSN and the subsequent character as a prefix.

TYPE = *ADDITIONAL-INFORMATION
An additional information request is sent to the administration entity. PROP-XT adds the TSN and the subsequent character (colon) as a prefix.

DATA = *NONE / <x-string 0..1000> / <c-string 0..500>
This operand specifies the message to be sent. An empty string is sent if *NONE is specified.

REFERENCE-NAME = *NO / <structured-name 3..30>
This operand identifies associated messages (message sequences with question / reply or command / result / additional information request / additional information / termination acknowledgment).

If a name is assigned by the administration procedure, it must not begin with the reserved character $ or #.

If a command (*CMD) is sent, the name specified here is used as the job ID. If *NO is specified, UCON has no job ID. If UCON requires a unique job ID for the command (this is the case if the command server permits overlapping operation), the first three characters of this name must be unique. This is because UCON uses the origin (identification of the administration entity connection) and the first three characters of the name for identification purposes.

A job event is defined with this name, i.e. it must be ensured that the name is also unique as an EVENT-NAME. The event belongs to the $$TRANSACTION group.

If you want to receive command results (IMPLICIT-EVENTS = *ALL or *CMD-TERMINATION), a REFERENCE-NAME must be specified.

In the case of responses to questions (*REPLY) and additional information (*ADDITIONAL-INFORMATION), the name transferred in <RESULT-VARIABLE>.REFERENCE-NAME on receipt of the question or additional information request, must be used here as the REFERENCE-NAME.

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A03	Internal errors	Resource bottleneck on system: Bourse error for main task.
0A1D		Port signals an error when sending over administration entity connection.
0A16		Error in the administration entity management when sending data to the administration entity.
0A17		No job could be generated for sending data to the administrator.
0A1D	Semantic errors	Error in connection to one or more administration entity: The connection to the administration entity is cleared.
0A1E		Not all specified administration entities are defined: The specified administration entity is not defined for the administration process.
0B0B		Although job elements are expected, no REFERENCE-NAME was given.
0B08		Specified REFERENCE-NAME already exists. Command ignored.
0E05		The user tried to send an empty string to an entity of the type DCAM or UTM. Empty strings may only be sent to entities of the OPERATING type.
010C		Syntax error in name.
0B1F	Warnings	An operand was specified which is only expedient in job processing (for example REFERENCE-NAME, IMPLICIT-EVENT). PROP-XT does not support jobs for this administration type (DCAM/UTM).
0B20		No message types are supported for entities of the type DCAM or UTM. The specification of TYPE is ignored.

Table 25: Error codes of the SEND-TO-PROP-OBJECT command

Example

Sending the broadcast message “Warning: SHUTDOWN in 10 minutes!” to all users:

```
/SEND-TO-PROP-OBJECT OBJECT-NAME=SYSTEM, -
/   TYPE=*CMD(IMPLICIT-EVENTS=*NO), -
/   DATA='/BCST ''Warning: SHUTDOWN in 10 minutes!'''
```

SIMULATE-PROP-EVENT

SIMULATE-PROP-EVENT can be used to simulate events, e.g. messages, commands, changes to job variable values, in order to test administration procedures. This command can also be used to initiate events directly (see also page 135).
The administration process in which SIMULATE-PROP-EVENT is entered and the administraton process for which an event is simulated, must run on the same processor.

(Section 1 of 2)

```
SIMULATE-PROP-EVENT

 PROCESS-NAME = *UNCHANGED / *OWN / list-poss(20): <structured-name 1..30>
,SELECT-EVENT = *FROM-OBJECT (...) / *FROM-ADMINISTRATOR(...) / *BY-EVENT-NAME(...) /
 *TIME-LIMIT-REACHED

    *FROM-OBJECT(...)
    |    OBJECT-NAME = <structured-name 1..30>
    |   ,EVENT-TYPE = *MSG / *QUESTION / *CMD-RESULT / *CMD-TERMINATION /
    |     *ADDITIONAL-INFORMATION-REQUEST
    |   ,EVENT-DATA = *SYSTEM-MSG(...) /  <c-string 1..1800> / <x-string 1..3600>
    |        *SYSTEM-MSG(...)
    |        |    MSG-ID = <name 7..7>
    |        |   ,INSERT-0 = *NO / <c-string 1..180>
    |        |   ,INSERT-1 = *NO / <c-string 1..180>
    |        |   ,INSERT-2 = *NO / <c-string 1..180>
    |        |   ,INSERT-3 = *NO / <c-string 1..180>
    |        |   ,INSERT-4 = *NO / <c-string 1..180>
    |        |   ,INSERT-5 = *NO / <c-string 1..180>
    |        |   ,INSERT-6 = *NO / <c-string 1..180>
    |        |   ,INSERT-7 = *NO / <c-string 1..180>
    |        |   ,INSERT-8 = *NO / <c-string 1..180>
    |        |   ,INSERT-9 = *NO / <c-string 1..180>
    |        |   ,INSERT-10 = *NO / <c-string 1..180>
    |        |   ,INSERT-11 = *NO / <c-string 1..180>
    |        |   ,INSERT-12 = *NO / <c-string 1..180>
    |        |   ,INSERT-13 = *NO / <c-string 1..180>
    |        |   ,INSERT-14 = *NO / <c-string 1..180>
    |   ,REFERENCE-NAME = *NO / <structured-name 3..30>
```

(Section 2 of 2)

```
*FROM-ADMINISTRATOR(...)
   |  EVENT-TYPE = *CMD / *REPLY / *ADDITIONAL-INFORMATION / *EXCHANGE-REQUEST
   | ,EVENT-DATA =  <c-string 1..500> / <x-string 1..1000>
   | ,REFERENCE-NAME = *NO / <structured-name 3..30>
*BY-EVENT-NAME(...)
   |  EVENT-NAME = <structured-name 1..30>
```

PROCESS-NAME =
This operand specifies the administration process to which the event is to be applied.

PROCESS-NAME = *UNCHANGED
The current SIMULATE-PROP-EVENT command applies to the same administration process as the previous SIMULATE-PROP-EVENT command issued from this administration process. If there is no previous SIMULATE-PROP-EVENT command, your own administration process is used, provided this is in test mode. If the administration process is not in test mode, the following message is output: “POP0A1A: Simulation is not possible for administration entities or events”.

PROCESS-NAME = *OWN
The simulated event is applied to your own administration process.

PROCESS-NAME = list-poss(20): <structured-name 1..30>
The simulated event is applied to the specified administration processes.

SELECT-EVENT =
This operand specifies how the simulated event is to be identified.

SELECT-EVENT = *FROM-OBJECT
This operand value specifies the administration entity from which the message originates.

OBJECT-NAME = <structured-name 1..30>

Name of the administration entity from which the message originates.

EVENT-TYPE = *MSG / *QUESTION / *CMD-RESULT / *CMD-TERMINATION / *ADDITIONAL- INFORMATION-REQUEST
Type of the specified message. The message type is entered as a key word.

EVENT-DATA = *SYSTEM-MSG (...) / <c-string 1..1800> / <x-string 1..3600>

This operand specifies whether the message is transferred in the form of a character string or as a message code with inserts. If the message is transferred as a character string, it begins at position 1 in <RESULT-VARIABLE>.INFORMATION. PROP-XT does **not** add a UCON header to the character string.

MSG-ID = <name 7..7>

Message code.

INSERT-i = *NO / <c-string 1..180>
INSERT-i identifies the message insert with the logical number i (where 0 ≤ i ≤ 14). The value *NO means that the value of insert i is not relevant.

i The combined length of all inserts must not exceed 180 characters.

REFERENCE-NAME = *NO / <structured-name 3..30>
This operand identifies associated messages (message sequences with question/reply or command/result/additional information request/additional information/termination acknowledgment) as such.
A name assigned by the administration procedure must not begin with the reserved characters $ or #.

SELECT-EVENT = *FROM-ADMINISTRATOR
This operand value specifies that an event originating from the administrator is simulated.

EVENT-TYPE = *CMD / *REPLY / *ADDITIONAL- INFORMATION
*CMD / *REPLY / *ADDITIONAL- INFORMATION: Type of the specified message. The message type is entered as a key word.
*EXCHANGE-REQUEST: The simulated event is a request to exchange an administration procedure.

EVENT-DATA = <c-string 1..500> / <x-string 1..1000>
For EVENT-TYPE = *CMD / *REPLY / *ADDITIONAL- INFORMATION, this operand specifies the message in the form of a character string.
For EVENT-TYPE = *EXCHANGE-REQUEST, this operand specifies the name of the new administraton procedure. SIMULATE-PROP-EVENT does not check the name for correct syntax.

In the case of ambiguous, abbreviated commands, PROP-XT cannot determine which command is called; however, the ambiguous abbreviation is **not** rejected.

i A command must not begin with a slash (/).

REFERENCE-NAME = *NO / <structured-name 3..30>
This operand identifies associated messages (message sequences with question/reply or command/result/additional information request/additional information/termination acknowledgment) as such.
A name assigned by the administration procedure must not begin with the reserved characters $ or #.

SELECT-EVENT = *BY-EVENT-NAME
This operand value specifies that the simulated event is taken directly from EVENT-NAME. This method is used particularly when simulating time events. If other types of event are transferred in this way, return variables referring to the content and type of the message are not supplied.

EVENT-NAME = <structured-name 1..30>
Event ID specified with START-PROP-EVENT-MONITORING.

SELECT-EVENT = *TIME-LIMIT- REACHED
This operand value simulates a TIMEOUT for a WAIT-FOR-PROP-EVENTS command in the administration procedure to be tested. Only one TIMEOUT can be simulated at a particular time. A new TIMEOUT cannot be simulated until this TIMEOUT has been retrieved.

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A0F	Syntax error	Total length of inserts too great.
0A03	Internal errors	Resource bottleneck on system: Access to initialized PROP-XT data is faulty.
0A16		Error in accessing administration entity management data.
0A17		Error in accessing event management data.
0A18		Error in accessing administration process management data.
0213		Specified system message could not be generated: Error in accessing the message file. Check whether this file actually contains the message with the specified MSG-ID.
0A1A	Semantic errors	Simulation not possible for administration entities or events.
0A10		Syntax error in command: The syntax of the command to be simulated is incorrect.
0A19	Warning	Event could not be simulated for some administration entities.

Table 26: Error codes of the SIMULATE-PROP-EVENT command

START-PROP-EVENT-MONITORING

This command is used to define events that are to be awaited with WAIT-FOR-PROP-EVENTS (see page 120ff). Incoming events are buffered by PROP-XT until the command is executed. The events that occurred are not transferred to the administration procedure until this command is issued.

The administration procedure can define how incoming events are to be buffered:

– All incoming events are buffered and are picked up by the administration process.

– Only the last event to arrive is picked up by the administration process. This means that a newer event overwrites one already buffered, if both are defined by the same START-PROP-EVENT-MONITORING.

Events are picked up in an administration procedure using the WAIT-FOR-PROP-EVENTS command and the identification of the desired events.

i Buffered events that are not picked up can cause buffer overflows. In this case, the error is reported with the next PROP-XT system command. If the buffer overflows, incoming events are lost.

PROP-XT offers a special handling facility for DADM messages: normally, a DADM message is split into numerous physical message segments. PROP-XT combines these message segments together so that a single WAIT command can be used to receive one logical DADM message.

If a message fulfils the criteria for two different event definitions, it is not generally the case with WAIT that the event first defined is the event first reported. The same applies to STD-ACTIONs.

START-PROP-EVENT-MONITORING can be specified at any point and any number of times in an administration procedure.

(Section 1 of 3)

START-PROP-EVENT-MONITORING
EVENT-NAME = <structured-name 1..30>
,**GR**OUP**-NAME** = ***NONE** / <structured-name 1..30>

(Section 2 of 3)

```
,SELECT-EVENT = *FROM-OBJECT (..) / *FROM-ADMINISTRATOR(..) / *TIMER(..) / *JV-MODIFICATION(..)

   *FROM-OBJECT(...)
        OBJECT-NAME = *ALL / list-poss(20): <structured-name 1..30>
       ,EVENT-DATA = *SYSTEM-MSG (...) / *FREE-STRING(...) / *ALL
          *SYSTEM-MSG(...)
               MSG-ID = <name 7..7>
              ,INSERT-0 = *NO / <c-string 1..180>
              ,INSERT-1 = *NO / <c-string 1..180>
              ,INSERT-2 = *NO / <c-string 1..180>
              ,INSERT-3 = *NO / <c-string 1..180>
              ,INSERT-4 = *NO / <c-string 1..180>
              ,INSERT-5 = *NO / <c-string 1..180>
              ,INSERT-6 = *NO / <c-string 1..180>
              ,INSERT-7 = *NO / <c-string 1..180>
              ,INSERT-8 = *NO / <c-string 1..180>
              ,INSERT-9 = *NO / <c-string 1..180>
              ,INSERT-10 = *NO / <c-string 1..180>
              ,INSERT-11 = *NO / <c-string 1..180>
              ,INSERT-12 = *NO / <c-string 1..180>
              ,INSERT-13 = *NO / <c-string 1..180>
              ,INSERT-14 = *NO / <c-string 1..180>
          *FREE-STRING(...)
               POSITION = 1 / <integer 1..1800>
              ,STRING = <c-string 1..1800>

   *FROM-ADMINISTRATOR(...)
        TYPE = *CMD (...) / *EXCHANGE-REQUEST
          *CMD (...)
               CMD-COMPLETITION = *NO / *YES
              ,CMD-INFO-REQUEST = *NO / *YES
              ,CMD-NAME = <structured-name 1..30>

   *TIMER(...)
        ZONE = *LOCAL-TIME (...) / *UTC (...)
          *LOCAL-TIME (...)
               DATE = *ANY / <date 8..10>
              ,TIME = *ANY / <time 1..8>
              ,SEASON = *WINTER-TIME / *SUMMER-TIME
          *UTC (...)
               DATE = *ANY / <date 8..10>
              ,TIME = *ANY / <time 1..8>
       ,INTERVAL = *NO / *DAILY / <integer 10..86400>
```

(Section 3 of 3)

```
    *JV-MODIFICATION(...)
     |   JV-NAME = <full-filename 1..54>
     |  ,POSITION = 1 / <integer 1..256>
     |  ,STRING = <c-string 1..256>
,BUFFER-EVENT = *ALL-NOT-PROCESSED / *LATEST-ONLY
```

EVENT-NAME = <structured-name 1..30>
This operand identifies an event definition. The specified name must be unique in the administration process. It must not begin with the reserved character $ or #.

GROUP-NAME = *NONE / <structured-name 1..30>
This operand specifies a group in which the event can be combined with other events. The group name must not begin with the reserved character $ or #. Event groups are defined implicitly by their first occurrence in a START-PROP-EVENT-MONITORING command, and make it easier to address events that can be logically combined. If *NONE is specified, the event is not assigned to any group.

SELECT-EVENT =
This operand specifies the type of event.

SELECT-EVENT = *FROM-OBJECT
This operand value specifies that a message from the administration entity results in an event.

Here you can only specify messages that are distributed via routing code, or job elements that belong to one of the commands issued by this administration process.

Whether a message is sent in relation to a command or via routing code depends on the command server and cannot be globally defined.

OBJECT-NAME = *ALL / list-poss(20): <structured-name 1..30>
Name of the administration entity. *ALL means that the message from any entity currently administered by the administration procedure results in an event.

EVENT-DATA =
Here you define which messages result in an event.

EVENT-DATA = *SYSTEM-MSG (...)
The message is a message with the specified MSG7 message code and the defined inserts.

MSG-ID=<name 7..7>
Message code.

INSERT-i = *NO / <c-string 1..180>
INSERT-i identifies the message insert with the logical number i (where 0 ≤ i ≤ 14). The value *NO means that the value of insert i is not used for comparison.

If a character string is specified here, it must be a defined insert value because only then will it be identified as an event.

ℹ The combined length of all inserts must not exceed 180 characters.

EVENT-DATA = *FREE-STRING (...)
A freely definable character string is compared.

POSITION = 1 / <integer 1..1800>
Position of the comparison string. The default value is the start of text (position1).

If it is a DADM message combined by PROP-XT from a number of message segments, it is necessary to specify the position in the combined message, i.e. the length of the preceding message segments must be added to the position in the message segment.

STRING = <c-string 1..1800>
Value of the comparison string.

EVENT-DATA = *ALL
All the messages of the specified administration object lead to an event.

Only use *ALL if you do not know the exact specification criteria. (Possibility of buffer overflow.)

SELECT-EVENT = *FROM-ADMINISTRATOR (...)
This operand value specifies that a command from the administrator results in an event.

TYPE =
This operand specifies whether a command or a request to exchange an administration procedure results in the event.

TYPE = *CMD (...)
A command results in the event.

CMD-COMPLETITION = *NO/*YES
This operand defines how the transferred command will be handled in the administration procedure. If *YES is specified, the administration procedure must send a command acknowledgment to complete the command. If *NO is specified, the command acknowledgment is sent by PROP-XT.
If no command acknowledgment is sent (*NO), no results can be sent either.

CMD-INFO-REQUEST =
This operand defines how the transferred command will be handled in the administration procedure. It is only possible to send requests for additional information if a command acknowledgment is also sent (CMD-COMPLETITION = *YES).

CMD-INFO-REQUEST = *NO
When a command is entered, the uniqueness of the command identification is not taken into account. This may make it impossible to forward the additional information request, because it can no longer be uniquely identified.

CMD-INFO-REQUEST = *YES
When a command is entered, the uniqueness of the command identification is taken into account. This ensures that the additional information request can be forwarded, because it can be uniquely identified.

CMD-NAME = <structured-name 1..30>
Name of the command that results in an event.

TYPE = *EXCHANGE-REQUEST
A request to exchange the administration procedure results in the event.

SELECT-EVENT = *TIMER
The event is a time event. Time events occur for the first time at the specified time (DATE, TIME). If an interval is defined, the event occurs again when the interval elapses.

If a concrete date is specified (DATE not equal to *ANY), but no specific time is specified (*ANY), the current time is selected.

If neither a particular time nor a concrete date is specified , the current date and time are selected, if an interval is specified. If no interval is specified, an error occurs.

If the specified time has already passed, the event definition is rejected as incorrect (regardless of the value of INTERVAL).

ZONE = *LOCAL-TIME (...)
You work with local time.

DATE = *ANY / <date 8..10>
Specifies the day used for calculating the time at which the event is to occur for the first time. In addition to DATE, this time is affected by the TIME specification. *ANY means that the date does not affect the time.

If no particular date is specified (*ANY) but a concrete time is specified (TIME not equal to *ANY), the current date is selected, if the time is still to occur on this day. If the time has already passed, the date of the following day is selected.

TIME = *ANY / <time 1..8>
Specifies the time in units of 10 seconds, which is to be used to calculate the time at which the event is to occur for the first time. In addition to TIME, this time is affected by the DATE specification. *ANY means that no particular time is defined.

SEASON = *WINTER-TIME / SUMMER-TIME
The operand SEASON ensures uninterrupted operation even when the time is switched over to winter time or summer time. You need only specify whether the specified time is to be interpreted as winter or summer time; all the other conversions that are required are performed automatically by PROP-XT.
SEASON is only evaluated if the time addressed is ambiguous or non-existent as a result of a changeover in time.

ZONE = *UTC (...)
You use Universal Coordinated Time.

DATE = *ANY / <date 8..10>
Specifies the day used for calculating the time at which the event is to occur for the first time. In addition to DATE, this time is affected by the TIME specification. *ANY means that the date does not affect the time.

If no particular date is specified (*ANY) but a concrete time is specified (TIME not equal to *ANY), the current date is selected, if the time is still to occur on this day. If the time has already passed, the date of the following day is selected.

TIME = *ANY / <time 1..8>
Specifies the time in units of 10 seconds, which is to be used to calculate the time at which the event is to occur for the first time. In addition to TIME, this time is affected by the DATE specification. *ANY means that no particular time is defined.

INTERVAL = *NO / *DAILY/ <integer 10..86400>
Defines an interval in seconds in which the time event occurs. The maximum interval is one day (86400 seconds).

*NO means that no periodic time event is desired.
Time events without an interval specification can only occur once and are therefore deleted after they have been picked up by the administration procedure. An explicit STOP-PROP-EVENT-MONITORING is thus not required and will be rejected.

*DAILY means that the event always occurs at the same time of day. This also applies on the days on which the changeover to summer or winter time occurs. If local time is used, the interval lasts 23 and 25 hours respectively.

SELECT-EVENT = *JV-MODIFICATION (...)
The event is the matching of a job variable with the specified value. Job variable events are not buffered. It is not checked whether the job variable has the desired value until a CHECK-PROP-EVENTS or WAIT-FOR-PROP-EVENTS refers to a job variable event. In this case, the event is transferred. Otherwise, the job variable is periodically polled with WAIT-FOR-PROP-EVENTS until it has the expected value (or until the time limit specified with WAIT-FOR-PROP-EVENTS expires).

JV-NAME = <full-filename 1..54>
Name of the job variable.

POSITION = 1 / <integer 1..256>
Position of the comparison string. The default value is the start of text (position 1).

STRING = <c-string 1..256>
Value of the comparison string. A distinction is made between uppercase and lowercase.

BUFFER-EVENT =
This operand defines whether all events not picked up are to be buffered, or only the last event.

BUFFER-EVENT = *ALL-NOT-PROCESSED
All events not picked up are buffered.

BUFFER-EVENT = *LATEST-ONLY
Only the last event belonging to a definition is buffered. Incoming events that are not buffered because *LATEST-ONLY is specified, are considered not to have occurred and are therefore not counted, e.g. with CHECK-PROP-EVENT-OCCURRENCE.

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
010C	Syntax error	Syntax error in name.
0A03	Internal errors	Resource bottleneck on system.
0A16		Error in accessing administration entity management data.
0A17		Error in accessing event management data.

Table 27: Error return codes of the START-PROP-EVENT-MONITORING command (Section 1 of 3)

MAINCODE	Error class	Meaning
0A0F	Semantic errors	Total length of inserts too great. The combined length of all the inserts in a system message may not exceed 180 characters.
0A04		Event type specified cannot be handled by PROP-XT.
0A15		Data inconsistency when processing system command.
0B0C		Event name already used.
0B0D		Event name already used as group name.
0B0E		Group name already used as event name.
0B09		Resource bottleneck with internal PROP-XT reference name: Too many commands and inquiries for which PROP-XT has assigned an internal reference name (max.: ~1300!) have not been terminated.
0B10		Permitted number of servers for operator special commands reached.
0B11		Connection error in signing on operator special command.
0B13		Time specified for time event already elapsed.
0B14		Illegal time specification.
0A1B	Warnings	No administration entities defined for administration process.
0A1D		Error in connection to one or more administration entity.
0A1E		Some of the administration entities specified for the administration process are not defined. Message “POP9A07: Administration entity ’(&01)’ not defined for administration process” is output for each administration entity. The message “POP9A06: Connection to administration entity ’(&01)’ not active” is output for each defined administration entity to which the connection is not active.
0240		The specified time falls within the double hour in the changeover from summer to winter time. A replacement time was determined on the basis of the SEASON operand.
0241		The calculated time falls within the double hour in the changeover from summer to winter time. A replacement time was determined on the basis of the SEASON operand.

Table 27: Error return codes of the START-PROP-EVENT-MONITORING command (Section 2 of 3)

MAINCODE	Error class	Meaning
0242		The time span with the specified time is omitted because of the changeover from winter to summer time. A replacement time was determined on the basis of the SEASON operand.

Table 27: Error return codes of the START-PROP-EVENT-MONITORING command (Section 3 of 3)

Example

Defining an event that enters all DUMP messages:

```
/START-PROP-EVENT-MONITORING EVENT-NAME=DUMP, -
/   SELECT-EVENT=*FROM-OBJECT(EVENT-DATA=SYSTEM-MSG -
/   (MSG-ID=IDA0N51))
```

START-PROP-OBJECT-MONITORING

This command defines an administration entity.

An administration entity can only be referenced after it has been defined. The command assigns a name to an administration entity. The administration entity can be of the type OPERATING, UTM or DCAM.

The connection to the administration entity is then established. Using different entity names, the administration of a number of administration entities by an administration process can be logically separated, even when the partner with which the connection is established remains the same.

If the connection cannot be established (e.g. because the entity cannot be reached or authorization is not available at the time), the entity is not known to PROP-XT as from this time. This also applies if a connection cannot be re-established successfully after it has been lost.
This makes it possible to define events that relate to this entity.

The loss of a connection is indicated to the administration process when an event is awaited from this entity or an attempt is made to send a message to the entity.

PROP-XT attempts to establish the connection again by issuing a new START-PROP-OBJECT-MONITORING command. Only the specifications of the new command are evaluated. PROP-XT does not issue a warning even if there are changes from the previous definition.

START-PROP-OBJECT-MONITORING

```
 OBJECT-NAME = <structured-name 1..30>
,OBJECT = *OPERATING (...) / *UTM-APPLICATION(...) / *DCAM-APPLICATION(...) / *FROM-DATABASE
    *OPERATING(...)
         USER-IDENTIFICATION = *OWN / <name 1..8>
        ,PASSWORD = *OWN / *NO / <c-string 1..8> / <x-string 1..16>
        ,PROCESSOR-NAME = *LOCAL / <name 1..8>
        ,OPERATOR-ROLE = *NO / <name 1..8>
    *UTM-APPLICATION(...)
         APPLICATION-NAME = <name 1..8>
        ,PASSWORD = *NO /  <c-string 1..4> / <x-string 1..8>
        ,PROCESSOR-NAME = *LOCAL / <name 1..8>
        ,OWN-APPLICATION = *BY-PROP / <name 1..8>
    *DCAM-APPLICATION(...)
         APPLICATION-NAME = <name 1..8>
        ,PASSWORD = *NO /  <c-string 1..4> / <x-string 1..8>
        ,LOGON-MSG = *NO /  <c-string 1..80> / <x-string 1..160>
        ,PROCESSOR-NAME = *LOCAL / <name 1..8>
        ,OWN-APPLICATION = *BY-PROP / <name 1..8>
```

OBJECT-NAME = <structured-name 1..30>
Name of the administration entity.

OBJECT = *OPERATING
The administration entity is of the *OPERATING type.

Access to the administration entity is via UCON. Authorization to enter commands and receive messages is linked to the user ID under which the administration process is running, and is transparent to PROP-XT.
The administration process behaves like an administrator that signs on dynamically to UCON. The procedure must therefore contain the corresponding REQUEST-OPERATOR-ROLE commands (see the System Operator’s Guide).

> **USER-IDENTIFICATION = *OWN / <name 1..8>**
> User ID for the administration entity.
> The specification *OWN causes a connection to be established only if the administration entity is on the local processor or on a processor which works with PROP-XT in an MSCF network of the type CCS. This remote processor must have at least BS2000 version 11.2.

PASSWORD = *OWN / *NO / <c-string 1..8> / <x-string 1..16>
Passwort for setting up a connection to the administration entity.

The specification *OWN causes a connection to be established only if the administration entity is on the local processor or on a processor which works with PROP-XT in an MSCF network of the type CCS. USER-IDENTIFICATION=*OWN must also be specified.

The specification *NO means that the user ID is not password protected.

In all other cases, the correct password for the user ID has to be specified.

PROCESSOR-NAME = *LOCAL / <name 1..8>
Name of the processor on which the administration entity is situated. The specification *LOCAL means that the administration entity is on a local processor.

OPERATOR-ROLE = *NO / <name 1..8>
If an OPERATOR-ROLE is specified, it is requested by PROP-XT after the connection has been set up.

OBJECT = *UTM-APPLICATION (...)
The administration entity is a UTM application.

APPLICATION-NAME = <name 1..8>
Name of the UTM application.

PASSWORD = *NO / <c-string 1..4> / <x-string 1..8>
Connection password for the UTM application.

PROCESSOR-NAME = *LOCAL / <name 1..8>
Name of the processor on which the UTM application runs. The specification *LOCAL means that the administration entity is on the local processor.

OWN-APPLICATION = *BY-PROP / <name 1..8>
Specifies the auxiliary application with which PROP-XT sets up the connection to the UTM application.

When *BY-PROP is specified, PROP-XT assigns a unique, PROP-XT-wide name for the auxiliary application. Since this name is not known when the KDCDEF for the UTM application is generated, the UTM administrator must permit connections in general to programs by specifying PTYP=APPLI in the TPOOL statement.

The name for OWN-APPLICATION must not start with the reserved character $; it must be unique. <name 1..8> must be specified in the PTERM statement for the KDCDEF of the UTM application.

OBJECT = *DCAM-APPLICATION (...)
The administration entity is a DCAM application.

APPLICATION-NAME = <name 1..8>
Name of the DCAM application.

PASSWORD = *NO / <c-string 1..4> / <x-string 1..8>
Connection password for the DCAM-Anwendung.

LOGON-MSG = *NO / <c-string 1..80> / <x-string 1..160>
Connection message transferred when the connection is set up.

PROCESSOR-NAME = *LOCAL / <name 1..8>
Name of the processor on which the DCAM application runs. The specification *LOCAL means that the administration entity is on a local processor.

OWN-APPLICATION = *BY-PROP / <name 1..8>
Specifies the auxiliary application with which PROP-XT sets up the connection to the DCAM application. The name for OWN-APPLICATION must not start with the reserved character $; it must be unique.
When *BY-PROP is specified, PROP-XT assigns a unique, PROP-XT-wide name for the auxiliary application.
The DCAM application can recognize an administration procedure by means of the name of OWN- APPLICATION.

OBJECT = *FROM-DATABASE
The information on the administration entity is to be retrieved by PROP-XT from the database.

Return codes

With entities of the type DCAM or UTM, the connection message sent by a partner is supplied to the variable SYSPOP_LOGON-MSG (string of length 80).

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Fehlerklasse	Bedeutung
0A30	Successful	The name must not start with "*" or "$".
0C00		No adminstration entity description exists for the administration entity in the database.
0C01		The user is not authorized to use this administration entity description from the database.

Table 28: Error return codes for the START-PROP-OBJECT-MONITORING command (Section 1 of 3)

MAINCODE	Fehlerklasse	Bedeutung
0A03	Internal errors	Resource bottleneck on system: Bourse error in communication between administration task and main task. No acknowledgment was received within 600 sec when setting up a connection to the administration entity.
0A1D		Error in connection to one or more administration entity: A negative acknowledgment was received when setting up the connection to the administration entity.
0A16		Error in accessing administration entity management data.
0E0C		A connection between OWN-APPLICATION and APPLICATION already exists.
0E0E		The auxiliary application specified under OWN-APPLICATION could not be opened.
0E0F		The connection to the administration entity could not be set up, since the auxiliary application OWN-APPLICATION has already been opened by another TASK exclusively or with password protection.
0E10		The partner or the partner’s transport system has rejected the connection set-up you requested. BCAM brings station information YDB_STCONREJ. Possible cause: incorrect password.
0E13		BCAM limits the number of connections an application may open. This number has been exceeded for the application specified under OWN-APPLICATION. Note: other tasks can also open connections from this application. If the error occurs with START-PROP-OBJ-MON <name>, type=*OPERATING, then connections to too many systems within the MSCF network of the type CCS have been set up throughout PROP without the specification of a user ID.

Table 28: Error return codes for the START-PROP-OBJECT-MONITORING command (Section 2 of 3)

MAINCODE	Fehlerklasse	Bedeutung
010A	Semantic errors	The connection to the administration entity already exists. The administration entity specified in the command has been defined earlier and the connection is active.
0107		The administration object is not of the type UCON.
0108		No futher administration processes/entities alllowed.
0E11		Partner application unknown.
0E12		Name of the partner system unknown.
0E14		The user has tried to set up a connection to an entity of the OPERATING type on a remote system without specifying a user ID. This will only function if the partner system and the local processor are in a common MSCF network of the type CCS, the network is active, and the BS2000 version of the partner system >= V11.2 ist.
0E15		Operator-Role is unknown or not permitted for this user ID.
0E16		The specified user ID or password is incorrect for the START-PROP-OBJECT-MONITORING command.

Table 28: Error return codes for the START-PROP-OBJECT-MONITORING command (Section 3 of 3)

Example

Defining the administration entity UCON:

```
/START-PROP-OBJECT-MONITORING OBJECT-NAME=UCON
```

STOP-PROP-EVENT-MONITORING

An event definition that was defined with START-PROP-EVENT-MONITORING (see page 110 ff.) is deleted with all buffered events. As of this point, PROP-XT no longer recognizes the event definition. This has the following consequences:

– Incoming events that correspond to the defined event definition are no longer delivered.

– None of the events that have already arrived for this event definition is delivered, rather all are discarded.

The STOP-PROP-EVENT-MONITORING command can be issued at any stage in an administration procedure. However, it is only possible to delete event definitions that were defined at an earlier point in the administration process and have not yet been deleted.

STOP-PROP-EVENT-MONITORING
EVENT-NAME = ***ALL** / list-poss(20): <structured-name 1..30>

EVENT-NAME = *ALL / list-poss(20): <structured-name 1..30>
This operand identifies event definitions. This can refer to an event definition defined with START-PROP-EVENT-MONITORING or a group definitions. *ALL means that all event definitions defined in this administration process are to be deleted.

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A03	Internal errors	Resource bottleneck on system: Bourse error in communication between administration task and main task.
0A17		Error in accessing event management data.
0B19	Semantic error	The events specified in the command were not defined beforehand.

Table 29: Error return codes of the STOP-PROP-EVENT-MONITORING command

Example

Deleting the event definition for the DUMP messages:

```
/STOP-PROP-EVENT-MONITORING EVENT-NAME=DUMP
```

STOP-PROP-OBJECT-MONITORING

An object defined with the START-PROP-OBJECT-MONITORING command (see page 110 ff.) is deleted with all buffered events.

This command is used by an administration process to indicate that it no longer administers an administration entity to which it previously signed on with a START-PROP-OBJECT-MONITORING command. This has the following consequences:

– The connection to the administration entity is cleared down.

– Buffered events caused by the administration entity are deleted and are not transferred to the administration process.

– Event definitions that refer to this entity are deleted.

STOP-PROP-OBJECT-MONITORING
OBJECT-NAME = *ALL / list-poss(20): <structured-name 1..30>

OBJECT-NAME = *ALL / list-poss(20): <structured-name 1..30>
This operand identifies an administration entity. *ALL means that all defined administration entities are to be deleted.

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A03	Internal error	Resource bottleneck on system. Bourse error in communication between administration task and main task.
0A16		Error in accessing administration entity management data.
0A17		Error in accessing event management data.
0A1B	Semantic error	No administration entities defined for administration process. NOTE: When *ALL is specified for the administration entities there is no check to see which ones have been defined.
0A1E	Warning	Not all specified administration entities are defined: Some of the administration entities specified for the administration process are not defined. Message “POP9A07: Administration entity ’(&01)’ not defined for administration process” is output for each defined administration entity.

Table 30: Error return codes of the STOP-PROP-OBJECT-MONITORING command

Example

Signing off the administration entity SYSTEM.

```
/STOP-PROP-OBJECT-MONITORING OBJECT-NAME=SYSTEM
```

WAIT-FOR-PROP-EVENTS

The administration procedure is shifted to a wait state with the WAIT-FOR-PROP-EVENTS command and is reactivated by:

- the occurrence of an event corresponding to the specified event definition. The event definition is defined with the START-PROP-EVENT-MONITORING command (see page 101ff)
- timeout

```
WAIT-FOR-PROP-EVENTS

 EVENT-NAME = *ALL / *SELECTED(...) / list-poss(20): <structured-name 1..30>

    *SELECTED(...)
          SELECTED-EVENTS = *TIMER / *JV-MODIFICATION / *FROM-ADMINISTRATOR /
             *FROM-OBJECT(...)
             *FROM-OBJECT(...)
                   OBJECT-NAME = *ALL / list-poss(20): <structured-name 1..30>

,TIME-LIMIT = 600 / *NO / <integer 0..86400>

,RESULT-VARIABLE = *STD-VARIABLE / <structured-name 1..20>
```

EVENT-NAME =
Specifies whether an event from all possible events, an event defined by SELECT-EVENTS, or an event specified here by its event ID is awaited.

EVENT-NAME = *ALL
An event from all possible events is awaited.

EVENT-NAME = *SELECTED (...)
Particular events are awaited. These must be selected as follows:

SELECTED-EVENTS =
This operand specifies the type of event awaited.

SELECTED-EVENTS = *TIMER
An time event is awaited.

SELECTED-EVENTS = *JV-MODIFICATION
An job variable event is awaited.

SELECTED-EVENTS = *FROM-ADMINISTRATOR
An input from the administrator is awaited.

SELECTED-EVENTS = *FROM-OBJECT (...)
This operand value defines the entity whose messages result in an event.

OBJECT-NAME=*ALL/ list-poss(20): <structured-name 1..30>
A message from all administration entities (*ALL) or only from particular entities specified here, results in an event.

EVENT-NAME = list-poss(20): <structured-name 1..30>
An event specified here by its event ID is awaited.

TIME-LIMIT = 600 / *NO / <integer 0..86400>
Wait time in seconds. The default value is 600 seconds. (The greatest possible accuracy in actual operation is one minute).

TIME-LIMIT = *NO
This operand value defines an unlimited wait time.

RESULT-VARIABLE = *STD-VARIABLE / <structured-name 1..20>
Defines whether results are to be transferred in the SYSPOP variable (*STD-VARIABLE) or in a variable specified here.

When a desired event occurs, a number of return codes are set (see section “Error handling” on page 79).

Return codes

Structure elements of the SYSPOP variable are assigned after the command is issued.

Apart from the general values specified on page 79 f., the following specific values can occur with errors:

MAINCODE	Error class	Meaning
0A0A	Semantic errors	Too many operands in command: The command defined as an event in the SDF-P variable cannot be processed since the number of operands is greater than 20.
0A0B		Command operands cannot be edited: The command defined as an event in the SDF-P variable cannot be edited since the value of the operand(s) is neither STRING nor INTEGER.
0A08		None of the specified events defined for administration process.
0A09		Invalid message received: Message received does not match the event type specified.
0A1B		None of the specified administration entities are defined for the administration process.
0A10		Syntax error in command: The syntax of the command received could not be analyzed by PROP-XT (SDF command analysis).
0A13		Arriving events can no longer be buffered for the administration process.
0225		Access to job variable not possible.
0A03	Internal errors	Resource bottleneck on system: Access to initialized PROP-XT data is faulty.
0A16		Error in accessing administration entity management data.
0A17		Error in accessing event management data.
020F		Subsystem JV not available.

Table 31: Error return codes of the WAIT-FOR-PROP-EVENTS command (Section 1 of 2)

MAINCODE	Error class	Meaning
0A1D	Warnings	Error in connection to one or more administration entity: The connection no longer exists to all the administration entities specified in the command. Message “POP9A06: Connection to administration entity ’(&01)’not active” is output for each such administration entity.
0A1E		Not all specified administration entities are defined: Some of the administration entities specified for the administration process are not defined. Message “POP9A07: Administration entity ’(&01)’ not defined for administration process” is output for each administration entity. Message “POP9A06: Connection to administration entity ’(&01)’ not active” is output for each defined administration entity to which the connection is not active.
0A14		Termination requested: Via the administration command SEND-TO-PROP-PROCESS the administration process was requested to terminate. Any wait time started can thus be terminated.
0B1B		Not all specified elements of list defined (no events occurred). The list elements are output with the message “POP9A04: Event ’(&01)’ not known”.
0B18		No events occurred for selection criterion: No more events have occurred for the specified events since those that have already been fetched/transferred, and the wait time for the WAIT command (TIME-LIMIT=) has elapsed. If OPERATIONAL-MODE=*SIMULATION applies for the administration process, an event generated by the SIMULATE-PROP-EVENT command (TIME-LIMIT-REACHED) aborts the WAIT command.

Table 31: Error return codes of the WAIT-FOR-PROP-EVENTS command (Section 2 of 2)

The structured variable <RESULT-VARIABLE> is defined as follows:

- <RESULT-VARIABLE>.EVENT-NAME:
  Event ID of the transferred event (TYPE = (STRING)).
- <RESULT-VARIABLE>.GROUP-NAME:
  Group ID of the transferred event (TYPE = (STRING)).
- <RESULT-VARIABLE>.EVENT-TYP:
  Type of the last event selected (TYPE = (STRING)).

  The following values are possible:

TIMER	Time event
JV	Job variable event
FROM-ADMINISTRATOR	Message from administrator
FROM-AO	Message from administration entity
ERROR	Error

Table 32: Possible values of the EVENT-TYP structure

- <RESULT-VARIABLE>.INFORMATION-TYP:
  Type of the last event selected (TYPE = (STRING)). This specification is not relevant to the EVENT TYPEs time and job variable events (nor errors) and is not defined in these cases.

  The following values are possible:

CMD	Command
REPLY	Reply
ADD-INFO	Additional information
MSG	Message
QUESTION	Question
CMD-RESULT	Command result
ADDITIONAL-INFORMATION-REQUEST	Additional information request
CMD-TERMINATION	Command termination acknowledgment
EXCHANGE-REQUEST	Exchange request

Table 33: Possible values of the INFORMATION-TYP structure

- <RESULT-VARIABLE>.PARTNER-NAME:
  Name of the partner that initiated the last event selected (TYPE = (STRING)).

Specifies the name of the administration entity from which the event originated. If this is a message from an administrator administered by UCON, the variable contains the value 'ADMINISTRATOR'.

- <RESULT-VARIABLE>.INFORMATION:
  This specification (TYPE = (STRING)) is not relevant to the EVENT TYPEs time and job variable events (nor errors) and is not defined in these cases. The incoming message is transferred in full (including UCON header).
- <RESULT-VARIABLE>.MES-ID:
  Message code, if a message in mapping format is received (TYPE = (STRING)).
- <RESULT-VARIABLE>.I*i* (*i*= 0 thru 14):
  Message insert *i*, if a message in mapping format is received which indicates an insert *i* (TYPE = (STRING)). If insert *i* is not contained in the message, a string of length 1 with the value ' ' is transferred.
- <RESULT-VARIABLE>.CMD-NAME:
  Command name, if a command is received (TYPE = (STRING)).
- <RESULT-VARIABLE>.OPERAND-*i* (*i* = 1 thru 20):
  Value of the *i*-th operand, if a command is received which indicates an operand *i*.

  If the operand is of type INTEGER, the SDF-P variable is also created with type INTEGER. In the case of character types, an SDF-P variable is created with type STRING.

  The following SDF operand types are not supported: REAL, X-STRING, LABEL, STRUCTURE, LIST, OR-LIST, X-TEXT, and FIXED. They result in an error during processing.

  The restriction also applies that commands that are first analyzed by SDF and then by PROP-XT are transferred to the administration procedure, have a linear SDF transfer range, and are limited to 20 operands.
- <RESULT-VARIABLE>.NEW-PROCEDURE:
  Value specified for the SEND-TO-PROP-PROCESS command in the NEW-PROCEDURE operand. This will usually be the name of the new procedure (TYPE = (STRING)).
- <RESULT-VARIABLE>.REFERENCE-NAME:
  Identification of a message sequence TYPE = (STRING)).

  If an event that occurred is described by several event definitions, the event is buffered a number of times and is transferred to the administration process with the corresponding WAIT command. This also applies to job-related, implicitly defined events that were defined with the SEND-TO-PROP-ADMINISTRATOR command using the REFERENCE-NAME operand.

Example

Synchronously awaiting all outstanding events:

```
/WAIT-FOR-PROP-EVENTS EVENT-NAME=*ALL, TIME-LIMIT=*NO
```

5.4 Sample administration procedure

This section contains a complete sample administration procedure, comprising the following:

- purpose of the procedure
- SDF-A statements for defining the user-defined commands contained in the administration procedure
- flowchart with references to the relevant lines in the procedure and CONSLOG log
- log of the entire administration procedure and the CONSLOG log created at runtime

Purpose

The administration procedure is to use the user-defined command

```
/ASSIGN-STD-PRINTER PRINTER=mn
```

to perform the following tasks:

- assign the printer identified by mn to BS2000 (ATTACH-DEVICE)
- assign the printer identified by mn to the SPOOL for the forms STD, RSLHP, and STDSF3 (SD)
- respond to message SPS0200 with TSN.Y

SDF-A statements

```
/START-PROGRAM $.SDF-A
OPEN-SYNTAX-FILE USERCOMMANDS.PROP-XT.V1.1A,TYPE=GROUP,MODE=CREATE
ADD-COMMAND ASSIGN-STD-PRINTER,IMPLEMENTOR=TPR(ENTRY=DUMMY,CALL=NEW)
ADD-OPERAND NAME=PRINTER,RESULT-OPERAND-NAME=*POSITION(POSITION=1)
ADD-VALUE TYPE=ALPHA-NAME(SHORTEST-LENGTH=2,LONGEST-LENGTH=4,WILDCARDS=NO)
END
```

Flowchart

```
                                                     |  Line no.    | Line no.
Action               (PROP-XT command)               |   Procedure  |  CONSLOG
________________________________________________________________________________

Sign on process to PROP-XT                           |     018      |
                     (BEGIN-PROP-PROCESS)            |              |
Sign on administration entity                        |     019      |  01 - 04
                     (START-PROP-OBJECT-MONITORING)  |              |
Request operator role from UCON                      |     021      |  05 - 08
                     (SEND-TO-PROP-OBJECT)           |              |
Sign on "ASSIGN-STD-PRINTER" as event                |     037      |  09 - 11
                     (START-PROP-EVENT-MONITORING)   |              |
WHILE-loop                                           |     042      |
   Wait for event                                    |     043      |
                     (WAIT-FOR-PROP-EVENTS)          |              |
   IF Returncode = OK                                |     044      |
      ATTACH mn                                      |     054      |  13 - 15
                     (SEND-TO-PROP-OBJECT)           |              |
      IF Returncode = OK                             |     084      |
         Assign printer to SPOOL   SD DEV=mn         |     093      |  16 - 18
                     (SEND-TO-PROP-OBJECT)           |              |
         IF Returncode = OK                          |     100      |
            OK acknowledgement to administrator      |     103      |  19
                     (SEND-TO-PROP-ADMINISTRATOR)    |              |
            Respond to MSG SPS0200                   |     109      |  24
                     (SEND-TO-PROP-OBJECT)           |              |
         ELSE (error in SD)                          |              |
            NOT OK acknowledgement to administrator  |     122      |
                     (SEND-TO-PROP-ADMINISTRATOR)    |              |
         END-IF                                      |              |
      ELSE (error in ATTACH)                         |     136      |
         NOT OK to Administrator                     |     146      |
                     (SEND-TO-PROP-ADMINISTRATOR)    |              |
      END-IF                                         |              |
   ELSE (abort job)                                  |              |
        Set abort condition for WHILE loop           |     156      |
   END-IF                                            |              |
END-WHILE                                            |     159      |
Sign off process from PROP-XT                        |     161      |
                     (END-PROP-PROCESS)              |              |
Error handling                                       |     165      |
```

Log of administration procedure

```
001 / SET-PROCEDURE-OPTIONS DATA-ESCAPE-CHAR=STD
002 / REMARK ************************ DOASP  *****************************
003 / REMARK *                                                           *
004 / REMARK * ATTACH  AND SD COMMAND FOR STANDARD PRINTER.             *
005 / REMARK * ACTIVATION VIA USER-DEFINED COMMAND                       *
006 / REMARK * "ASSIGN-STD-PRINTER PRINTER=MN"                           *
007 / REMARK *                                                           *
008 / REMARK *************************************************************
009 / DECLARE-VARIABLE NAME=SYSPOP(TYPE=STRUCTURE)
010 /                                               "RETURN VARIABLE"
011 / DECLARE-VARIABLE NAME=CONTINUE(TYPE=BOOLEAN,INIT-VALUE=TRUE)
012 /                                               "CONTROL OF WHILE LOOP"
013 / DECLARE-VARIABLE NAME=VSDRC(TYPE=BOOLEAN,INIT-VALUE=TRUE)
014 /                                               "RC OF SDVC COMMAND"
015 / DECLARE-VARIABLE NAME=VATTRC(TYPE=BOOLEAN,INIT-VALUE=TRUE)
016 /                                               "RC OF ATTACH COMMAND"
017 / BEGIN-BLOCK                                   "START OF COMMAND BLOCK"
018 / BEGIN-PROP-PROCESS PROCESS-NAME=DOASP         "SIGN ON PROCESS TO PROP-XT"
019 / START-PROP-OBJECT-MONITORING -
020 /    OBJECT-NAME=DOASP-OBJECT001                "SIGN ON AO"
021 / SEND-TO-PROP-OBJECT -
022 /    OBJECT-NAME=DOASP-OBJECT001, -
023 /    TYPE=CMD(IMPLICIT-EVENTS=CMD-TERMINATION), -
024 /    DATA='REQ-OPER-ROLE ROLEALL', -
025 /    REFERENCE-NAME=REQOPERROLE                 "REQUEST OPERATOR ROLE FROM UCON"
026 / WAIT-FOR-PROP-EVENTS -
027 /    EVENT-NAME=REQOPERROLE                     "WAIT FOR ACKNOWLEDGMENT"
028 / IF CONDITION=(SYSPOP.MAINCODE<>'0000')
029 /    GOTO LABEL=FEHLER                          "ABORT IN EVENT OF TIMEOUT"
030 / ELSE-IF CONDITION=(SYSPOP.MES-ID<>'NBR0740') -
031 /    OR (SYSPOP.I3<>CMD0001')                   "CHECK COMMAND RESULT"
032 /    MAINCODE=SYSPOP.I3
033 /    MELDUNG=MSG(MSG-ID='&MAINCODE')
034 /    SH-VAR (MAINCODE,MELDUNG)
035 /    GOTO LABEL=PROZENDE                        "ABORT IN EVENT OF NEGATIVE RC"
036 / END-IF
037 / START-PROP-EVENT-MONITORING -
038 /    EVENT-NAME=ASP, -
039 /    SELECT-EVENT=FROM-ADM -
040 /    (CMD-COMPLETITION=*YES, -
041 /    CMD-NAME=ASSIGN-STD-PRINTER)               "COMMAND AS RESULT IN PROP-XT"
```

```
/ WHILE (CONTINUE=TRUE)                            "LOOP OF ALL COMMAND JOBS"
/    WAIT-FOR-PROP-EVENTS E-N=ASP,T-L=NO           "WAIT FOR COMMAND"
/    IF (SYSPOP.MAINCODE='0000')                   "COMMAND ARRIVED"
/       SET-VAR REFNAM=SYSPOP.REFERENCE-NAME       "NOTE REF. NAME FOR TERM. ACK."
/       SET-VAR DEVMN=SYSPOP.OPERAND-1             "NOTE MN"
/       START-PROP-EVENT-MONITORING -
/          EVENT-NAME=NKR0041, -
/          SELECT-EVENT=(FROM-OBJECT -
/          (OBJECT-NAME=DOASP-OBJECT001, -
/          EVENT-DATA=SYSTEM-MSG -
/          (MSG-ID=NKR0041, -                      "EVENT "NKR0041" (MESSAGE TEXT)
/          INSERT-0=C'&DEVMN'))                    "SIGN ON"
/       SEND-T-PROP-OBJECT -
/          OBJECT-NAME=DOASP-OBJECT001, -
/          TYPE=CMD(IMPLICIT-EVENTS=CMD-TERMINATION),-
/          DATA=C'ATTACH-DEVICE &DEVMN', -
/          REF-N=ATT                               "ATTACH COMMAND"
/       W-F-P-E E-N=ATT,TIME-LIMIT=55              "WAIT ACKNOWLEDGMENT FOR ATTACH"
/       IF (SYSPOP.MAINCODE='0000') -
/          AND (SYSPOP.MES-ID='NBR0740') -
/          AND (SYSPOP.I3='CMD0001')               "ATTACH EXECUTED SUCCESSFULLY"
/          STOP-P-EVENT-MON E-N=NKR0041            "SIGN OFF EVENT NKR0041"
/          SET-VAR VATTRC=TRUE                     "AND RC=OK "
/       ELSE-IF CONDITION=(SYSPOP.MES-ID='NBR0740') -
/          AND (SYSPOP.I3='NKR0044') -
/          AND (SYSPOP.MAINCODE='0000')            "IF ATTACH REJECTED"
/          W-F-P-E NKR0041,TIME-LIMIT=0            "CHECK WHETHER PRINTER ALREADY ATTACHED"
/          IF (SYSPOP.MAINCODE='0000')             "DEVICE ALREADY ATTACHED"
/             STOP-P-EVENT-MON E-N=NKR0041         "SIGN OFF EVENT AGAIN"
/             SET-VAR VATTRC=TRUE                  "AND RC=OK "
/          ELSE
/             SET-VAR VATTRC=FALSE                 "SHOW ERROR IN ATTACH"
/             ATTMSG='NKR0044'                     "NOTE NKR0044 FOR ACKNOWLEDGMENT"
/          END-IF
/       ELSE-IF CONDITION=(SYSPOP.MAINCODE<>'0000')
/          SET-VAR VATTRC=FALSE                    "SHOW ERROR IN ATTACH"
/          ATTMSG=SYSPOP.MAINCODE                  "NOTE MSG FOR ACKNOWLEDGMENT"
/          CONTINUE=FALSE                          "END LOOP"
/       ELSE
/          SET-VAR VATTRC=FALSE                    "SHOW ERROR IN ATTACH"
/          ATTMSG=SYSPOP.I3                        "NOTE MSG FOR ACKNOWLEDGMENT"
/       END-IF
/       IF (VATTRC=TRUE)

/          REMARK ** IF ATTACH OK, ASSIGN PRINTER **

/          START-PROP-EVENT-MONITORING -
/             EVENT-NAME=SPS0200, -
/             SELECT-EVENT=FROM-OBJECT -
/             (OBJECT-NAME=DOASP-OBJECT001, -
/             EVENT-DATA=SYSTEM-MSG -
/             (MSG-ID=SPS0200, -
/             INSERT-1=C'&DEVMN'))               "MSG SPS0200 IS EVENT"
/          SEND-T-PROP-OBJECT -
/             OBJECT-NAME=DOASP-OBJECT001, -
/             TYPE=CMD(IMPLICIT-EVENTS=CMD-TERMINATION), -
/             DATA=C'SD DEV=&DEVMN, -
/             FORM=(STD,RSLHP,STDSF3)', -
/             REF-N=SDVC                         "ASSIGN PRINTER TO SPOOL"
/          W-F-P-E E-N=SDVC,TIME-LIMIT=55        "WAIT FOR ACKNOWLEDGMENT"
/          IF CONDITION=(SYSPOP.MES-ID='NBR0740') -
/             AND (SYSPOP.I3='CMD0001') -
/             AND (SYSPOP.MAINCODE='0000')       "CHECK RESULT"
/             SEND-T-P-A -
/                TYPE=CMD-TERM(), -
/                DATA=C'CMD0001', -              "OKAY ACKNOWLEDGMENT FOR A-S-P"
/                REF-N=&REFNAM                   "SEND TO ADMINISTRATOR"
/             W-F-P-E E-N=SPS0200,T-L=55         "WAIT FOR MSG SPS0200"
/             IF (SYSPOP.MAINCODE='0000')        "IF EVENT OCCURS:"
/                SEND-T-P-O -
/                   O-N=DOASP-OBJECT001, -
/                   TYPE=REPLY, -
/                   DATA=C'Y', -
/                   REFERENCE-NAME=&(SYSPOP.REFERENCE-NAME)
/                                                "RESPOND TO MESSAGE WITH 'Y'"
/             END-IF
/             STOP-P-E-M E-N=SPS0200             "SIGN OFF EVENT"
/          ELSE-IF CONDITION=(SYSPOP.MAINCODE='0000')
/                                                "NEGATIVE TERMINATION MESSAGE"

/             REMARK ** SDVC-FEHLER **

/             SDMSG=SYSPOP.I3                    "NOTE MSG# FOR ACKNOWLEDGMENT"
/             STOP-P-EVENT-MON E-N=SPS0200       "SIGN OFF EVENT"
/             SEND-T-P-A -
/                TYPE=CMD-TERM -
/                (RETCO-SUBCODE2=WARNING), -
/                DATA=C'&SDMSG', -
/                REF-N=&REFNAM                   "SEND ERROR ACK. TO ADMINISTRATOR"
/          ELSE                                  "NO TERMINATION MESSAGE RECEIVED"
/             STOP-P-EVENT-MON E-N=SPS0200       "SIGN OFF EVENT"
/             SEND-T-P-A -
/                TYPE=CMD-TERM -
/                (RETCO-SUBCODE1=INTERNAL), -
/                DATA=C'&(SYSPOP.MAINCODE)', -   "FOR EVENT A-S-P:SYSPOP.MAINCODE<>0000"
/                REF-N=&REFNAM                   "SEND ERROR ACK. TO ADMINISTRATOR"
/             SET-VAR CONTINUE=FALSE             "ABORT CONDITION FOR"
/          END-IF                                "SET WHILE LOOP"
/       ELSE

/          REMARK ** ATTFEHLER **

/          STOP-P-EVENT-MON E-N=NKR0041          "SIGN OFF EVENT"
/          IF CONDITION=(CONTINUE=FALSE)
/             SEND-T-P-A -
/                TYPE=CMD-TERM -
/                (RETCO-SUBCODE1=INTERNAL), -
/                DATA=C'&ATTMSG', -
/                REF-N=&REFNAM                   "SEND COMMMAND RETURN CODE"
/          ELSE
/             SEND-T-P-A -
/                TYPE=CMD-TERM -
/                (RETCO-SUBCODE2=WARNING), -
/                DATA=C'&ATTMSG', -
/                REF-N=&REFNAM                   "SEND COMMMAND RETURN CODE"
/          END-IF
/       END-IF                                   "WAIT FOR NEW JOB"
/    ELSE

/       REMARK ** SOMEONE SENT THE 'SEND-T-P-PROCESS' COMMAND **

/       SET-VAR SYSPOPMC=SYSPOP.MAINCODE         "SAVE RC FOR MSG TO CONSOLE"
/       SET-VAR CONTINUE=FALSE                   "ABORT CONDITION FOR"
/                                                "SET WHILE LOOP"
/    END-IF
/ END-WHILE                                      "END OF LOOP"

/ REMARK ** TERMINATION OF PROCEDURE DUE TO SYSPOP-MAINCODE<>0000 **

/ END-PROP-PROCESS                               "SIGN OFF PROCESS FROM PROP-XT"
/ IF CMD-ERROR
/ END-IF
/ END-BLOCK                                      "END OF COMMAND BLOCK"
/ IF-BLOCK-ERROR
/   GOTO LABEL=FEHLER                            "HANDLE PROCEDURE ERROR"
/ ELSE

/    REMARK ******** NO ERROR OCCURRED ********

/    GOTO LABEL=PROZENDE
/ END-IF
```

```
/FEHLER:

/ REMARK ******** ERROR OCCURRED ********

/ MAINCODE=MAINCODE(); SUBCODE1=SUBCODE1(); SUBCODE2=SUBCODE2()
/ MELDUNG=MSG(MSG-ID='&MAINCODE')                "MESSAGE TEXT"
/ SH-VAR (MAINCODE,SUBCODE1,SUBCODE2,MELDUNG)

/ REMARK ******** TERMINATE ADMINISTRATION PROCESS ********

/ END-PROP-PROCESS                               "SIGN OFF PROCESS"
/PROZENDE:

/ REMARK ******** END OF PROCEDURE ********

/ EXIT-PROCEDURE
```

CONSLOG log

```
01 OPRT /@@01-#09.112916 GRANT-PROP-CONNECTION STATION-NAME='$POP0009',TIME-
   LIMIT=100
02 @@01 ! UCO-#09.112916  %  NBR0740 COMMAND COMPLETED 'GRANT-PROP-CONNECTION'
   ; (RESULT: SC2=000, SC1=000, MC=CMD0001)
03   <E %XAA6-000.112916  %  NBR0797 APPLICATION '@@06' CONNECTED WITH
   '$CONSOLE', PROCESSOR NAME 'D015ZE24',
04 STATION NAME '$POP0009'
05 OPRT /@@06-REQ.112916 REQ-OPER-ROLE ROLEALL
06   <* % UCO-000.112916  %  NBR0970 OPERATOR TASK WITH TSN 'XAA7' CREATED FOR
   CONSOLE '@@06'
07 @@06 +XAA7-REQ.112916  %  NBR0980 OPERATOR ROLE 'ROLEALL' ASSIGNED TO
   OPERATOR ID 'PF1EISNE'
08 @@06 ! UCO-REQ.112916  %  NBR0740 COMMAND COMPLETED 'REQ-OPER-ROLE' ;
   (RESULT: SC2=000, SC1=000, MC=CMD0001)
09 OPRT /@@01-$AA.112917 CONNECT-CMD-SERVER CMD=PARAMETERS(CMD-NAME=ASSIGN-STD
   -PRINTER              ,
10 AUTHORIZATION-CODE='r'),SELECT=EXTERNAL-SERVER (COMPLETION
   -CONTROL=YES(QUESTION-POSSIBLE=NO ))
11 @@01 ! UCO-$AA.112917  %  NBR0740 COMMAND COMPLETED 'CONNECT-CMD-SERVER' ;
   (RESULT: SC2=000, SC1=000, MC=CMD0001)
12 @@01 /@@04-CM1.113002 ASSIGN-STD-P LW
13 OPRT /@@06-ATT.113002 ATTACH-DEVICE LW
14   <G % MSG-000.113002  %  NKR0040 'DEVICE     =LW' ATTACHED
15 @@06 ! UCO-ATT.113004  %  NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE' ;
   (RESULT: SC2=000, SC1=000, MC=CMD0001)
16 OPRT /@@06-SDV.113004 SD DEV=LW,FORM=(STD,RSLHP,STDSF3)
17 @@06 +XAA7-SDV.113005  %  SPS0B06 SPOOLOUT COMMAND PROCESSING COMPLETED FOR
   DEVICE 'LW'
18 @@06 ! UCO-SDV.113005  %  NBR0740 COMMAND COMPLETED 'SD' ; (RESULT: SC2=000,
   SC1=000, MC=CMD0001)
19 @@04 ! UCO-CM1.113005  %  NBR0740 COMMAND COMPLETED 'ASSIGN-STD-P' ;
   (RESULT: SC2=000, SC1=000, MC=CMD0001)
20   <S %   S-000.113006  %  SPS0404 DEVICE 'LW' ACTIVATED. FOB SIZE: 000000;
   CHARACTER SETS 000004; ROTATION:
21  'NO'; FO-OV: 'ANY'
22   <S ?SPAA-000.113006  %  SPS0200 TSN '0IKI': FORM 'STD' MOUNTED ON PRINTER
   'LW'? REPLY (Y=YES; N=NO; C,<FO
23 RMNAME>=NEW FORM)?
24 SPAA R@@06-000.113006.Y
25   <S % UCO-000.113006  %  NBR0792 QUESTION '000' FROM 'SPAA' ANSWERED BY
   '@@06'. REPLY: Y
26   <S %SPAA-000.113022  %  SPS0120 TSN '0IKI', DEVICE 'LW': SPOOLOUT
   COMPLETED FOR FILE ':V11T:$TSOS.BULLETIN'
```

5.5 Test system

PROP-XT provides users with a test system for testing the administration procedures they create. The main advantages of this are listed below:

- Commands sent to administration entities are not executed.
- Administration entities need not be real.
- Events, such as messages, commands, and job variable modifications, can be created by the tester using the SIMULATE-PROP-EVENT command (see page 97). This command can also be used to directly initiate events.
- Tests are easier to automate.

The administration process is started as in real operation, but the operand OPERATIONAL-MODE=*SIMULATION must be specified with the BEGIN-PROP-PROCESS command (see page 81 ff). The commands contained in the process are handled in different ways:

- SDF-P and BS2000 commands are executed independently of PROP-XT, e.g. a command such as “/ERASE <filename>” is executed.
- With START-PROP-OBJECT-MONITORING (see page 110 ff), no connection is set up to the administration entity.
- SEND-TO-PROP-OBJECT (see page 93 ff) and SEND-TO-PROP-ADMINISTRATOR (see page 88 ff) are simulated. Their return information is output to the SYSOUT device assigned to the process.
- When simulating SEND-TO-PROP-ADMINISTRATOR, it is not checked:
  - whether REFERENCE-NAME is unique
  - whether the administration process is authorized to use REFERENCE-NAME
  - whether the job type is compatible with the current job status
- With SIMULATE-PROP-EVENT the following is not checked:
  - whether REFERENCE-NAME is unique and exists,
  - whether the job type is compatible with the current job status
- The commands START-PROP-EVENT-MONITORING (see page 101 ff) and STOP-PROP-EVENT-MONITORING (see page 116 ff) are executed as in real operation. The commands WAIT-FOR-PROP-EVENTS (see page 120 ff) and CHECK-PROP-EVENT-OCCURRENCE (see page 83 ff) only affect events that were created with SIMULATE-PROP-EVENT, i.e. the times and values of job variables are not relevant for time events and job variable events.

All simulated events are sent in the order in which they occur. In real mode, however, job variable events are always sent first with WAIT-FOR-PROP-EVENTS.

The SIMULATE-PROP-EVENT command is entered by the test station, and simulates an event. Like the other PROP-XT system commands, this command can be entered in a dialog or be contained in an administration procedure. This can be the (user-defined) procedure to be tested, or a remote procedure running in another task.
The administration process in which SIMULATE-PROP-EVENT is entered and the administraton process for which an event is simulated, must run on the same processor.

This command does not include an authorization check.

The SIMULATE-PROP-EVENT command simulates an event that is awaited using WAIT-FOR-PROP-EVENTS (see page 120 ff). PROP-XT buffers this event until the WAIT-FOR-PROP-EVENTS command is invoked.

Example 1: SIMULATE-PROP-EVENT in the procedure to be tested

Once the test phase is concluded, the SIMULATE-PROP-EVENT command and the OPERATIONAL-MODE=*SIMULATION operand must be removed from the procedure.

```
/SET-PROC-OPTIONS
     .
     .
     .
/BEGIN-PROP-PROCESS PROCESS-NAME=usercheck, -
/   OPERATIONAL-MODE=*SIMULATION
/START-PROP-OBJECT-MONITORING OBJECT-NAME=system, -
/   OBJECT=*OPERATING
     .
/START-PROP-EVENT-MONITORING EVENT-NAME=logon, -
/   SELECT-EVENT=*FROM-OBJECT -
/   (OBJECT-NAME=system, -
/   EVENT-DATA=*SYSTEM-MSG(MSG-ID=JMS0066))
     .
     .
     .
/REMARK   Simulation of event
/SIMULATE-PROP-EVENT PROCESS-NAME=*OWN, -
/   SELECT-EVENT=*FROM-OBJECT -
/   (OBJECT-NAME=system, -
/   EVENT-TYPE=*MSG, -
/   EVENT-DATA=*SYSTEM-MSG(MSG-ID=JMS0066,INSERT-0=’name-x’,INSERT-3=’ABCD’))
/REMARK  From this point onwards, the event is contained in the buffer
/WAIT-FOR-PROP-EVENTS -
/   EVENT-NAME=logon
/IF CONDITION=(SYSPOP.I0 EQ ...)
/   SEND-TO-PROP-OBJECT OBJECT-NAME=system -
/      TYPE=*CMD, -
/      DATA='/CANCEL-JOB JOB-ID=TSN(&(SYSPOP.I3), -
/      REFERENCE-NAME=can
/END-IF
```

Screen output:

```
/CANCEL........
```

Example 2: SIMULATE-PROP-EVENT in your own test procedure

Once the test phase is concluded, the OPERATIONAL-MODE=*SIMULATION operand must be removed from the procedure. The process in task 2 could also run with OPERATIONAL-MODE=*OPERATION.

```
   Task 1                                          Task 2

/SET-PROC-OPTIONS                       |    /SET-PROC-OPTIONS
     .                                  |
     .                                  |
/BEGIN-PROP-PROCESS  -                  |    /BEGIN-PROP-PROCESS -
/   PROCESS-NAME=usercheck, -           |    /   PROCESS-NAME=test, -
/   OPERATIONAL-MODE=*SIMULATION        |    /   OPERATIONAL-MODE=*SIMULATION
/START-PROP-OBJECT-MONITORING -         |
/   OBJECT-NAME=system, -               |
/   OBJECT=*OPERATING                   |
     .                                  |
/START-PROP-EVENT-MON EVENT-NAME=logon,-|
/   SELECT-EVENT=*FROM-OBJECT -         |
/   (OBJECT-NAME=system, -              |
/   EVENT-DATA=*SYSTEM-MSG -            |
/   MSG-ID=JMS0066))                    |
     .                                  |
/WAIT-FOR-PROP-EVENTS -                 |
/   EVENT-NAME=logon                    |
     .                                  |   /REMARK Simulation of event
/REMARK Wait state is canceled         <--  /SIMULATE-PROP-EVENT -
                                        |   /   PROCESS-NAME=usercheck, -
     .                                  |   /   SELECT-EVENT=*FROM-OBJECT, -
                                        |   /   (OBJECT-NAME=system, -
/IF CONDITION=(SYSPOP.IO EQ..)          |   /   EVENT-TYPE=*MSG, -
                                        |   /   EVENT-DATA=*SYSTEM-MSG -
/   SEND-TO-PROP-OBJECT -               |   /   (MSG-ID=JMS0066,
/      OBJECT-NAME=system, -            |   /   INSERT-0=...,
/      TYPE=*CMD, -                     |   /   INSERT-3=...,))
/      DATA='/CANCEL-JOB....', -        |
/      REFERENCE-NAME=can               |
/END-IF                                 |
```

Screen output:

```
/CANCEL ........                        |
```

6 User-defined commands

PROP-XT user-defined commands are processed by administration processes. They must be defined by the programmer with SDF-A and must sign on to PROP-XT with START-PROP-EVENT-MONITORING. PROP-XT forwards this sign-on request to UCON.

An administration procedure signs on for processing user-defined commands with START-PROP-EVENT-MONITORING.

In the sign-on request, the administration procedure defines which protocol is to be used for command processing, i.e. whether command bracketing is supported, and whether overlapping command processing is permitted.

If the user-defined command is entered by an administrator, the command at the central wait point (WAIT-FOR-PROP-EVENTS) is transferred to the administration procedure, provided the authorization check is successful (the administrator is authorized to enter the command, and the administration procedure is signed on for command processing, see page 34).

If command bracketing is used, it is possible to send additional information requests (SEND-TO-PROP-ADMINISTRATOR) and receive additional information (WAIT-FOR-PROP-EVENTS). In this case, the termination acknowledgment must be sent from the administration procedure to the administrator (SEND-TO-PROP-ADMINISTRATOR).

If an administration procedure signs on for processing a command for which another administration procedure has already signed on, the command is transferred as of this point to the administration procedure that last signed on. From the point of view of UCON, the administration procedure acts as a command server for operator special commands: if four command servers have already signed on for the command, any attempt of a new command server to sign on will be rejected.

If user-defined commands are used, it must be ensured that their syntax is checked both in the SDF environment of the command originator and in the processing SDF environment (ID under which the administration process is running). This means that the command description must be incorporated into the group syntax file of the command originator and of the executing ID, see also the section “Syntax files“, (page 143).

With the WAIT-FOR-PROP-EVENTS command, the restrictions described for command events apply to user-defined commands in accordance with processing:

– User-defined commands have a linear SDF transfer range.

– User-defined commands have a maximum of 20 operands.

– The operand types REAL, X-STRING, LABEL, STRUCTURE, LIST, OR-LIST, X-TEXT, FIXED are not supported. These result in processing errors.

7 Generation and startup

This chapter describes subsystem generation and the creation of generation data for PROP-XT. It also contains information on the syntax files, and statements for starting and terminating PROP-XT.

7.1 DSSM generation

The file SYSSSC.PROP-XT.011 must be included with SSCM in the subsystem catalog (see the Subsystem Management manual):

7.2 PROP-XT generation

PROP-XT generation data can be divided into two groups:

- data for setting general operating parameters with MODIFY-PROP-PARAMETERS, which are predefined by PROP-XT with default values
- data for controlling the STD-ACTION standard action. The STANDARD-ACTION-AO administration entity, from which PROP-XT constantly receives messages as initiating events, must be described using CREATE-PROP-OBJECT-DESCRIPTION.

If the job variable $TSOS..SYSSSI.PROP-XT.011 does not yet exist, it is cataloged in the PROP-XT initialization phase. As soon as PROP-XT can be fully administered, this job variable is supplied with message number POP9900, and the associated message is output. Similarly, when terminating the PROP-XT subsystem, the job variable is supplied with message number POP9500.

Generation data is transferred to PROP-XT by means of administration commands. It is therefore possible to modify generation values during runtime.

By predefining the operands of the MODIFY-PROP-PARAMETERS command accordingly, you can start up a functional PROP-XT subsystem without the previous generation.

The remaining data required can be transferred to PROP-XT using the administration commands CREATE-PROP-OBJECT-DESCRIPTION and DELETE-PROP-OBJECT-DESCRIPTION. These commands can also be issued from administration procedures using SEND-TO-PROP-OBJECT.

If the generation data is to apply for the current session only, it should be incorporated using administration commands that are processed during runtime. If the data is modified, the operands REPLACE-OLD-ACTION and REPLACE-OLD-OBJECT must be used.

If the generation data is to apply beyond the current session, the corresponding administration commands must be stored in a command file. This can be created and modified using editors. The command file can be executed at any time during runtime using the command /RUN. It provides a simple means of globally modifying generation data.

The database, which contains the generation data, is thus available only at runtime.

7.3 Syntax files

The syntax of PROP-XT system commands and administration commands is defined either in the system syntax file or in the subsystem syntax file (SYSSDF.PROP-XT.011). The section “Authorization checks” on page 34 describes the privileges necessary to execute these commands.

In accordance with SDF logic for special users or user groups, the system administrator can use group syntax files to extend or restrict the set of functions and commands (see the System Administrator’s Guide or Introductory Guide to Systems Support).

When using user-defined commands, you must ensure that the command syntax is checked in the SDF environment of both the originator and the executor, i.e. the user ID under which the administration process runs. The command description must therefore be entered in the group syntax file of both the originator (an administrator that cannot be assigned a user ID works under the ID TSOS) and in the group syntax file of the executor ID.
The command description must contain the specification IMPLEMENTOR=TPR(ENTRY=DUMMY,CALL=NEW).

To define the syntax of a user-defined command, proceed as follows:

- describe the syntax in a group syntax file (SDF-A).
- assign PROFILE-IDs to the user IDs involved as SDF group IDs in the commands ADD-USER and MODIFY-USER.
- assign the group syntax file to the PROFILE-ID of the user ID using the MODIFY-SDF-PARAMETERS command.

7.4 Starting and terminating

The PROP-XT subsystem defined before BS2000 STARTUP in the DSSM catalog (as described in section “DSSM generation” on page 141) is started using the START-SUBSYSTEM command, and terminated using the STOP-SUBSYSTEM command.

8 SYSTEMDUMP MANAGER - a PROP-XT application

Using the PROP-XT application SYSTEMDUMP MANAGER (SDM) allows you to control and administer system dumps more easily. You can specify whether dumps are only to be created if certain conditions are met. Such a condition could be that a dump is created only when a system error occurs for the fifth time. The file names of the system dumps that occur are stored permanently, allowing you to access dumps from earlier sessions quickly and easily.

The SYSTEMDUMP MANAGER is a useful tool for analyzing error situations, and can also be used as an example for programming a PROP-XT application with the language resources of SDF-P. In view of this, the following describes not only how the procedure is used, but also how it works. If you are not interested in programming the application and only want to use it, you should skip the section “Functional description“.

8.1 Starting and terminating the SYSTEMDUMP MANAGER

The administration process is started using the command:

```
/ENTER-PROCEDURE *LIB-ELEM(SYSPRC.PROP-XT.011,SDM),
PROCEDURE-PARAMETER=(...)
```

The process signs on to PROP-XT using the name SDM.

The administration process is terminated either with the PROP-XT administration command

```
/SEND-TO-PROP-PROCESS PROCESS-NAME=SDM
```

or by answering the question POPAAAA to the operator with "N" (see page 148).

A third option is to abort the administration process with the BS2000 CANCEL command.

In all three cases, the following message is sent to the administrator:

```
POPAAA1     SYSTEMDUMP MANAGER terminated (MC=(&00))
```

After termination, you can resume using the information from the previous session stored in the HISTORY FILE by restarting the administration process with the same parameters.

This means that it is also possible to resume using information from HISTORY-FILEs from even earlier sessions.

The following values can be specified in the operand PROCEDURE-PARAMETER:

`OPERATOR-ROLE = list-poss(10):<name 1..8>`

A request is made to assign the specified OPERATOR-ROLEs.

`DUMP-LIMIT-PER-HOUR = <integer 0..255>`

Maximum number of system errors to be handled in an hour.

Standardwert: 5

`DUMP-OCCURRENCE = *ANY / <integer 1..255>`

This operand specifies after what number of occurrences of an address-specific system error a dump is to be created.

Default: 1

HISTORY-FILE = <full-filename 1..54>

File of the type SAM in which dump-specific data is stored by the administration process.

Default: SDM.HISTORY

MODUL-LIST-FILE = <full-filename 1..54>

File in which combined outputs (module names) are stored by the user.

Default: SDM.MODUL-LIST

None of the files specified as PROCEDURE-PARAMETER needs to exist.

If a PROCEDURE-PARAMETER is incorrectly specified, the procedure is terminated after SYSOUT with one of the following messages:

```
POPAAAB     DUMP-LIMIT-PER-HOUR not in defined range of values
            (<integer 0..255>)

POPAAAC     DUMP-OCCURRENCE not in defined range of values
            (<integer 1..255> or *ANY)

POPAAAD     HISTORY-FILE not a SAM file

POPAAAE     Problems signing on to PROP-XT subsystem

POPAAAF     OPERATOR-ROLE not specified or incorrect

POPAAA3     MODUL-LIST-FILE not a SAM or ISAM file (KEY-LEN=8)
```

8.2 Functional description

Automatic dump handling is performed by the SYSTEMDUMP MANAGER until a global dump counter has exceeded the number of dumps specified in an hour (DUMP-LIMIT-PER-HOUR).

The global dump counter records the number of system dumps that occured and were handled in the last hour. It is updated every 10 minutes by a timer function.

If the DUMP-LIMIT-PER-HOUR has been exceeded, the operator is asked how to proceed when the next system error occurs:

```
POPAAAA     Current DUMP-LIMIT-PER-HOUR ((&00)) of SYSTEMDUMP
            MANAGERs reached or exceeded by (&01)! Continue? Response:
            Y/N
```

– If the question is answered with "Y", this system error is handled by the administration process, despite DUMP-LIMIT-PER-HOUR having been exceeded.

– If the question is answered with "N", this means that it no longer makes sense to use the administration process. The SYSTEMDUMP MANAGER is terminated in this case.

When the system is waiting for the operator response, the messages IDA0N48 and IDA0N54, as well as all the user commands of the SYSTEMDUMP MANAGER can be received and their functions executed.

The date of the last occurence of a dump, the number of system errors that occurred at this point, and the file name of the last dump created are stored in the HISTORY-FILE for each dump address.

A dump address mostly consists of a hexadecimal error address with the structure <module name+nnnnn>.

There are two ways of handling error situations in the system programs of the BS2000: Each error situation is handled at the point it occurs, or several error situations are handled in a special module to which the system branches when an error occcurs (“combined output”). There are accordingly differences in the address of the dump that occurs.

Method 1: Dump address is not a “combined output”

This is the normal case. Each error situation is handled at the point it occurs (i.e. in the module).

A dump is created if DUMP-OCCURRENCE = *ANY or if no dump file exists in the HISTORY-FILE for this dump address and the value of DUMP-OCCURRENCE has been reached or exceeded.

Method 2: Dump address is a “combined output”

In many products of the system it is usual to always skip to a certain module of the product and to create a dump there when an error occurs. This results in errors with different causes always occurring at the same address “combined output”. It is therefore necessary in such cases to create a dump each time an error occurs.

You are therefore provided with the option of storing as many module names as you like in a MODUL-LIST-FILE, with each module name having to be written on a separate line.

If a module name of a dump address is then contained in the MODUL-LIST-FILE, a dump is always created for this dump address.

System dump handling cycle

The occurrence of a system error is detected by means of one of the messages IDA0N50 or IDA0N51. The address is transferred in the insert (&00) and can, for example, have the following features:

1F6E0382(YFJOSF2@+01772) or

1EDE1C76(YNDACOM+00396), EC=48,ELSN=0000000A,CODE=NTR0200

If one of these messages is received, the global dump counter is incremented and a check performed to determine whether the value of DUMP-LIMIT-PER-HOUR (dumps per hour) has been exceeded.

If it has, the operator is asked how to proceed with the message POPAAAA.

– If the question is answered with "Y", this system error is handled by the administration process, despite DUMP-LIMIT-PER-HOUR having been exceeded.

– If the question is answered with "N", this means that it no longer makes sense to use the administration process. The SYSTEMDUMP MANAGER is terminated in this case.

It is possible here to adapt the DUMP-LIMIT-PER-HOUR to a changed situation using the command MODIFY-SDM-PARAMETERS.

If the DUMP-LIMIT-PER-HOUR has not been reached, an attempt is made to determine the TSN from the message, in order to store the address of the system error in a TSN-specific SDF-P variable. The address of the system error is determined from the insert (&00). If the address for the system error cannot be determined from the insert (&00), the message is output after SYSOUT for diagnosis purposes. A dump of this type cannot be handled by the SYSTEMDUMP MANAGER.

Thereafter, the adddress of the system error is compared with existing dump addresses (ADDRESS) in the HISTORY-FILE:

- If the address already exists, the associated date (DATE) is updated and the counter (COUNT) incremented by 1.
- If the address does not already exist, it is included in the HISTORY-FILE together with the date and counter.

A dump has to be created if one of the following conditions applies:

- The setting DUMP-OCCURRENCE = *ANY applies.
- The dump-specific counter has reached the value of DUMP-OCCURRENCE.
- The dump-specific counter has exceeded the value of DUMP-OCCURRENCE and there is no dump file with the name DUMP-FILE. This means that a system dump cannot be created properly.
- A module name specified by the user in the MODUL-LIST-FILE occurs in the address of the system error (“combined output”).

Answering the query IDA0N52

When the following question is received:

```
IDA0N52     SYSTEM DUMP DESIRED? REPLY (EOT=DISK;<VOLUMETYPE>=TAPE;
            N=NO)
```

the response is given with the TSN-specific SDF-P variable after the TSN has been determined.

- If the varibale has the value "N", handling of this dump is terminated and the affected TSN-specific SDF-P variables are both deleted.
- If the variable has the value "Y", the message IDA0N54 has to be awaited before the dump file name can be transferred.

Entering dump file names in the HISTORY-FILE

When the message:

```
IDA0N54     '(&00)' WRITTEN TO FILE '(&01)'
```

appears, the address of the system error (stored in the TSN-specific SDF-P variable) is searched for in the HISTORY-FILE after the TSN has been determined, and the name of the dump file that is specified in the insert (&01) of the message is added. The two TSN-specific SDF-P variables are then deleted.

When the message

```
IDA0N48     TASK/SYSTEM SETTINGS PROHIBIT DUMP
```

appears, the associated system error entry in the HISTORY-FILE and the two TSN-specific SDF-P variables are deleted after determination of the TSN.

If the TSN-specific SDF-P variables are not found for the messages IDA0N52, IDA0N54 and IDA0N48 die TSN-spezifischen, none of the above-mentioned responses to the message in question can be performed, and there is no error message after SYSOUT.

8.3 Commands

The commands are user-defined commands according to PROP-XT.

SDM indicates command termination with a command termination message.

SHOW-SDM-PARAMETERS

This command outputs the current values of the control quantities and names of the HISTORY-FILE and MODUL-LIST-FILE.

SHOW-SDM-PARAMETERS

This command does not have any operands.

Example

```
/SHOW-SDM-PARAMETERS
DUMP-LIMIT-PER-HOUR=5
DUMP-OCCURRENCE=1
HISTORY-FILE=SDM.HISTORY
MODUL-LIST-FILE=SDM.MODUL-LIST
```

SHOW-SDM-HISTORY-FILE

This command outputs the entries in the HISTORY-FILE.

SHOW-SDM-HISTORY-FILE

This command does not have any operands.

Example

Output of the HISTORY-FILE (2 entries), where the dump file has not been created for the first entry and the dump for the second entry last occurred on 01.02.1994 and has already been created:

```
/SHOW-SDM-HISTORY-FILE

ADDRESS=1F6E0382(YFJOSF2@+01772)

   DATE=1994-02-02

   COUNT=1

   DUMP-FILE=

ADDRESS=1EDE1C76(YNDACOM+00396)

   DATUM=1994-02-01

   COUNT=5

   DUMP-FILE=$SYSDUMP.YNDACOM.00396.48.FJ01.D931203.140730
```

SHOW-SDM-MODUL-LIST-FILE

This command outputs the entries of the MODUL-LIST-FILE.

SHOW-SDM-MODUL-LIST-FILE

This command does not have any operands.

Example

Output of a MODUL-LIST-FILE with eight entries.

```
/SHOW-SDM-MODUL-LIST-FILE
DSSYS
DSYSSTART
DKCMAIN
NRT
DQPAM
NLMLOCK2
CMDESPL@
DHSBMSG@
```

MODIFY-SDM-PARAMETERS

This command can be used to change the control quantities and names of the HISTORY-FILE and MODUL-LIST-FILE.

MODIFY-SDM-PARAMETERS
DUMP-LIMIT-PER-HOUR = *UNCHANGED / <integer 0..255> **DUMP-OCCURRENCE** = *UNCHANGED / *ANY / <integer 0..255> **HISTORY-FILE** = *UNCHANGED / <full-filename 1..54> **MODUL-LIST-FILE** = *UNCHANGED / <full-filename 1..54>

DUMP-LIMIT-PER-HOUR = *UNCHANGED / <integer 0..255>

This operand specifies the maximum number of system errors handled in an hour.

DUMP-OCCURRENCE = *UNCHANGED / *ANY / <integer 0..255>

This operand specifies after what number of occurrences of an address-specific system error a dump is to be created.

HISTORY-FILE = *UNCHANGED / <full-filename 1..54>

Name of the file in which dump-specific data is stored. This file must be of the type SAM.

MODUL-LIST-FILE = *UNCHANGED / <full-filename 1..54>

Name of the file in which combined outputs (module names) are stored by the user. The file must be of the type SAM or ISAM (KEY-LEN=8).

If the HISTORY-FILE and/or MODUL-LIST-FILE has been exchanged without it being possible to terminate the cycle of one or more dump handlings, it is not possible to forecast how the dumps in question are to be handled.

Example

The DUMP-LIMIT-PER-HOUR is change to 9.

```
/MODIFY-SDM-PARAMETERS DUMP-LIMIT-PER-HOUR=9
```

8.4 Generation

The system application SYSTEMDUMP MANAGER (SDM) comprises an SDF-P administration procedure SDM, which is stored in the library SYSPRC.PROP-XT.011 as a library element of the type J.

The SYSTEMDUMP MANAGER builds on the PROP-XT subsystem. This means that PROP-XT must have been installed and started.

8.5 Authorization

The ID under which the administration process runs must have the privilege PROP-ADMINISTRATION (see section 2.10 Authorization checks).
The authorization (routing code) is also required to receive the messages IDA0N48, IDA0N50, IDA0N51, IDA0N52 and IDA0N54.

9 Messages

```
POPAAAA    Current DUMP-LIMIT-PER-HOUR ((&00)) of SYSTEMDUMP-MANAGERs reached or exceeded
           by (&01)! Continue? Response:Y/N
```

Warranty: Y (B) Routing code: R Weight: 50

Meaning
If you want this system error to be handled by the
SYSTEMDUMP-MANAGER despite the fact that DUMP-LIMIT-PER-HOUR
has been exceeded, respond with "Y".
You can also adjust DUMP-LIMIT-PER-HOUR to a changed situation
using the MODIFY-SDM-PARAMETER command.
If it no longer makes sense to use the SYSTEMDUMP-MANAGER, terminate,
the administration process by responding with "N"
Until this question has been answered, no new dump messages can
be processed by the SYSTEMDUMP-MANAGER.

```
POPAAAB    DUMP-LIMIT-PER-HOUR not in defined range of values (<integer 0..255>)
```

Warranty: Y

Response
Correct and restart the administration process.

```
POPAAAC    DUMP-OCCURRENCE not in defined range of values (<integer 1..255> or *ANY)
```

Warranty: Y

Response
Correct and restart the administration process.

```
POPAAAD    HISTORY-FILE not a SAM file. Command not executed
```

Warranty: Y (B) Routing code: R Weight: 50

```
POPAAAE    Problems signing on to PROP-XT subsystem
```

Warranty: Y

Meaning
The SYSTEMDUMP-MANAGER may only be used once per system or
PROP-XT is not installed.

Response
Terminate any administration process that is running and restart the
SYSTEMDUMP-MANAGER using ENTER-PROCEDURE or install PROP-XT.

POPAAAF `OPERATOR-ROLE not specified or incorrect`
Warranty: Y

Response
Correct and restart the administration process.

POPAAA1 `SYSTEMDUMP MANAGER terminated (MC=(&00))`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
SYSTEMDUMP MANAGER was terminated.
Cause: See maincode MC (PROP-XT User Guide).

POPAAA2 `SYSTEMDUMP MANAGER working with empty HISTORY-FILE`
Warranty: Y (B) Routing code: R Weight: 50

POPAAA3 `MODUL-LIST-FILE not a SAM or ISAM file (KEY-LEN=8). Command not executed`
Warranty: Y (B) Routing code: R Weight: 50

POPAAA4 `SYSTEMDUMP MANAGER working with empty MODUL-LIST-FILE`
Warranty: Y (B) Routing code: R Weight: 50

POP0A0A `Too many operands in command`
Warranty: Y

Meaning
The command defined as an event in the SDF-P variable
cannot be processed since the number of operands is greater than 20.

POP0A0B `Command operands cannot be edited`
Warranty: Y

Meaning
The command defined as an event in the SDF-P variable
cannot be edited since the value of the operand(s) is neither
STRING nor INTEGER.

POP0A0C `/BEGIN-PROP-PROCESS must be first PROP-XT command in an administration procedure.`
Warranty: Y

POP0A0D `Specified command is not a PROP-XT system command`
Warranty: Y

Meaning
The command received could not be identified as a PROP-XT
system command on account of the internal command name.
It might be a PROP-XT administration command, but if so
it may not be used in an administration procedure.

POP0A0E No such PROP-XT system command in syntax file
Warranty: Y

Meaning
In the syntax file a command was marked as an PROP-XT system command although it is not one.

POP0A0F Total length of inserts too great
Warranty: Y

Meaning
The combined length of all the inserts in a system message may not exceed 180 characters.

POP0A00 No PROP-XT authorization
Warranty: Y

Meaning
The user ID which was used to try to sign on to PROP-XT does not have the PROP-ADMINISTRATION privilege.

POP0A01 Administration process with this name has already signed on
Warranty: Y

POP0A02 Memory bottleneck
Warranty: Y

Meaning
The message buffer (MTBUFF) for communicating with the main task cannot be created.

POP0A03 Resource bottleneck on system
Warranty: Y

Meaning
GENERAL
Access to initialized PROP-XT data is faulty.

BEGIN-PROP-PROCESS
Bourse error in communication between administration task and main task.
System error in querying the user ID of the administration process.
System error in querying the TSN of the administration process.

END-PROP-PROCESS
Bourse error in communication between administration task and main task.
Error in releasing AP name for NAME-MANAGER.
Error in releasing communication buffer (MTBUFF).

START-PROP-EVENT-MONITORING
Bourse error in communication between administration task and main task. No acknowledgment was received within 600 sec when signing on a command. A negative acknowledgment was received from UCON when signing on a command as an event.

STOP-PROP-EVENT-MONITORING, STOP-PROP-OBJECT-MONITORING
Bourse error in communication between administration task and main task. An internal error has occurred.

START-PROP-OBJECT-MONITORING
Bourse error in communication between administration task and main task. No acknowledgment was received withihin 600 sec when setting up a connection to the administration entity.

SEND-TO-PROP-ADMINISTRATOR, SIMULATE-PROP-EVENT
Bourse error in communication between administration task and main task.

SEND-TO-PROP-OBJECT
Bourse error for main task.

`POP0A04` `Event type specified cannot be handled by PROP-XT`
Warranty: Y

`POP0A08` `None of the specified events defined for administration process`
Warranty: Y

Meaning
No events that can be waited for have been defined by,
the administration process.
The events specified in the list are not defined.

`POP0A09` `Invalid message received`
Warranty: Y

Meaning
Message received does not match the event type specified.

`POP0A1A` `Simulation not possible for administration entities or events`
Warranty: Y

```
POP0A1B    No administration entities defined for administration process
```

Warranty: Y

Meaning

WAIT-FOR-PROP-EVENTS, START-PROP-EVENT-MONITORING
None of the specified administration entities are defined
for the administration process.

STOP-PROP-OBJECT-MONITORING
None of the specified administration entities are defined
for the administration process.
NOTE: When *ALL is specified for the administration entities
there is no check to see which ones have been defined.

```
POP0A1C    Port signals error when sending over administration connection
```

Warranty: Y

```
POP0A1D    Error in connection to one or more administration entities
```

Warranty: Y

Meaning

WAIT-FOR-PROP-EVENT, START-PROP-EVENT-MONITORING
The connection no longer exists to all the administration
entities specified in the command.
Message POP9A06 is output for each such administration entity.

START-PROP-OBJECT-MONITORING
A negative acknowledgment was received when setting up the
connection to the administration entity.

SEND-TO-PROP-OBJECT
The connection to the administration entity
specified by /SEND-TO-PROP-OBJECT is cleared.
Port signals an error when sending over administration entity
connection.

```
POP0A1E    Not all specified administration entities are defined
```

Warranty: Y

Meaning

WAIT-FOR-PROP-EVENTS, START-PROP-EVENT-MONITORING
Some of the administration entities specified for the
administration process are not defined.
Message POP9A07 is output for each administration entity.
Message POP9A06 is output for each defined administration
entity to which the connection is not active.

STOP-PROP-OBJECT-MONITORING
Some of the specified administration entities are not defined for the administration process.
Message POP9A07 is output for each administration entity.

SEND-TO-PROP-OBJECT
The specified administration entity is not defined for the administration process.

POP0A1F `Not all specified events are defined`
Warranty: Y

Meaning
The list of specified events includes some, which are not defined.
Message POP9A04 is output for each such event.

POP0A10 `Syntax error in command`
Warranty: Y

Meaning
GENERAL
The syntax of the command received could not be analyzed by PROP-XT (SDF command analysis).

SIMULATE-PROP-EVENT
The syntax of the command to be simulated is incorrect.

POP0A11 `Command allowed again only after END-PROP-PROCESS`
Warranty: Y

Meaning
A second BEGIN-PROP-PROCESS command is only permitted in an administration task after an END-PROP-PROCESS command.

POP0A13 `Arriving events can no longer be buffered for the administration process`
Warranty: Y

Meaning
The memory space required exceeds the value permitted by the administration command MODIFY-PROP-PARAMETERS.

POP0A14 `Termination requested`
Warranty: Y

Meaning
Via the administration command SEND-TO-PROP-PROCESS the administration process was requested to terminate.
WAIT-FOR-PROP-EVENT
Via the Administration command SEND-TO-PROP-PROCESS the administration process was requested to terminate. Any wait time started can thus be terminated.

POP0A15 `Data inconsistency when processing system command`
Warranty: Y

Response
- Create fault documentation
- notify system service

POP0A16 `Error in accessing administration entity management data`
Warranty: Y

Meaning
GENERAL
Error in accessing the administration entity management data.

SEND-TO-PROP-OBJECT
Error in the administration entity management when sending data to the administration entity.

POP0A17 `Error in accessing event management data`
Warranty: Y

Meaning
GENERAL
Error in accessing event management data.

START-PROP-EVENT-MONITORING
No acknowledgment was received within 600 sec when signing on a command. A negative acknowledgment was received from UCON when signing on a command as an event.

SEND-TO-PROP-OBJECT
No job could be generated for sending data to the administrator.

POP0A18 `Error in accessing administration process management data`
Warranty: Y

POP0A19 `Event could not be simulated for some administration entities`
Warranty: Y

POP0A20 `Subsystem PROP-XT stopped`
Warranty: Y

POP0A21 `If a user ID is specified, a password must also be given.`
Warranty: Y (B) Routing code: R Weight: 50

`POP0A30    Illegal Value for OWN-APPLICATION`

Warranty: Y

Meaning

The name may not start with "*" or "$".

Response

Correct and try again.

`POP0B0A    Unknown REFERENCE-NAME`

Warranty: Y

Meaning

The user has specified a REFERENCE-NAME that does not identify an order in this administration process.
The order may already have been terminated.

Response

Check the REFERENCE-NAME.

`POP0B0B    Although order elements are expected, no REFERENCE-NAME was given`

Warranty: Y

Response

Specify REFERENCE-NAME.

`POP0B0C    Event name already used`

Warranty: Y

`POP0B0D    Event name already used as group name`

Warranty: Y

`POP0B0E    Group name already used as event name`

Warranty: Y

`POP0B00    Illegal event type`

Warranty: Y

`POP0B01    Illegal data type`

Warranty: Y

`POP0B08    Specified REFERENCE-NAME already exists. Command ignored.`

Warranty: Y

Meaning

The specified REFERENCE-NAME already exists. The command was ignored.
or was not unambiguous in the specified length.
When sending queries and commands, the REFERENCE-NAME
must be unambiguous in the first 3 characters.

Response

Choose another REFERENCE-NAME.

POP0B09 `Resource bottleneck with internal PROP-XT reference name`
Warranty: Y

Meaning
Too many commands and inquiries for which PROP-XT has assigned an internal reference name (max.: >1300!) have not been terminated.

Response
Check the administration procedures for endless loops.

POP0B1B `Not all specified elements of list defined.`
Warranty: Y

Meaning
In case of wait or check event: No events occurred

POP0B1C `Command server no longer connected`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
The command event is no longer defined in the server administration process.

POP0B1D `Command could not be transferred to server administration process`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Buffer overflow owing to problems with server administration process.

POP0B1F `PROP-XT does not support jobs for this entity type. Command will be executed`
Warranty: Y

Meaning
An operand was specified that is only relevant for job processing (e.g. REFERENCE-NAME, IMPLICIT-EVENT).
However, PROP-XT does not support any jobs with this administration type (DCAM/UTM).

POP0B10 `Permitted number of servers for operator special commands reached`
Warranty: Y

POP0B11 `Connection error in signing on operator special command`
Warranty: Y

`POP0B12` `Message type currently not allowed`
Warranty: Y

Meaning
The order identified by the specified REFERENCE-NAME
is in a status in which the specified message type is not allowed.

Response
Check the REFERENCE-NAME, message type and status of the order.

`POP0B13` `Time specified for time event already elapsed`
Warranty: Y

`POP0B14` `Illegal time specification`
Warranty: Y

`POP0B18` `No events occurred for selection criterion`
Warranty: Y

Meaning
WAIT-FOR-PROP-EVENT
No more events have occurred for the specified events since those that have already been fetched/transferred, and the wait time for the WAIT command (TIME-LIMIT=) has elapsed.
If OPERATIONAL-MODE=SIMULATION applies for the administration, process, an event generated by the SIMULATE-PROP-EVENT command (TIME-LIMIT-REACHED) aborts the WAIT command.

CHECK-PROP-EVENT-OCCURRENCE
No more events have occurred for the specified events since those that have already been fetched/transferred.

`POP0B19` `The events specified in the command were not defined beforehand`
Warranty: Y

`POP0B20` `Message type not supported.`
Warranty: Y

Meaning
For entities of type DCAM or UTM no message types are supported.
The TYPE Specification is ignored.

POP0C00 `Administration entity description not in database`
Warranty: Y

Meaning
No administration entity description is contained in the database for the administration entity.

Response
The attributes of the administration entity must be supplied with START-PROP-OBJECT-MONITORING or the administration entity must be defined in the database by the PROP-XT administrator.

POP0C01 `No authorization to use the administration entity description from the database`
Warranty: Y

Meaning
The user is not authorized to use this administration entity description from the database.

Response
The PROP-XT administrator must authorize this user (user ID) to use this administration entity description from the database.

POP0E0C `Connection already exists`
Warranty: Y

Meaning
A connection already exists between OWN-APPLICATION and APPLICATION

Response
Select a different name for OWN-APPLICATION and try again

POP0E0E `OWN-APPLICATION cannot be opened.`
Warranty: Y

Meaning
The auxiliary application specified under OWN-APPLICATION could not be opened.
Possible reasons:
- Application is pregenerated and protected by a user password
- BCAM problems such as
-     shortage of memory space
-     threshold values exceeded
-     name reserves or AP ID exhausted
- BCAM shutdown announced

Response
Retry command later.

POP0E0F `Application specified in OWN-APPLICATION already open`
Warranty: Y

Meaning
The connection to the application entity could not be set up because the auxiliary application OWN-APPLICATION has already been opened exclusively or with password protection by another task.

Response
Specify a different name for OWN-APPLICATION.

POP0E05 `No data specified.`
Warranty: Y

Meaning
The user attempted to send an empty string to an entity of the type DCAM or UTM. Empty strings may only be sent to entities of the type Operating.

POP0E10 `Connection rejected by partner.`
Warranty: Y

Meaning
The partner or his transport system rejected the connection request. BCAM issues Station Information YDB_STCONREJ.
Possible reason: incorrect password

POP0E11 `Partner Application not known`
Warranty: Y

Response
Correct and try again.

POP0E12 `Name of partner system not known`
Warranty: Y

Response
Correct and try again

POP0E13 `Maximum number of connections per application exceeded.`
Warranty: Y

Meaning
BCAM limits the number of connections that an application may open.
This number has been exceeded for the application specified under OWN-APPLICATION.
Note: Other tasks can also open connections from this application.

If an error occurs with START-PROP-OBJ-MON <name>, type=*OPERATING too many connections within PROP have already been set up to computers within the user’s own MSCF network (of the type CCS) without a user ID being specified.

Response
With type = DCAM/UTM: select a different value for APPLICATION.
With type = OPERATING: notify the system administrator.

POP0E14 `Computer not in common MSCF network`
Warranty: Y

Meaning
The user attempted to set up a connection to an entity of the type Operating on a foreign system without specifying the user ID. This works only if the partner system and the local system are both in a common MSCF network of the type CCS, the network is active and the BS2000 version used on the partner system is >= V11.2.

Response
Notify the system administrator or specify an operator user ID.

POP0E15 `Illegal operator role`
Warranty: Y

Meaning
Operator role unknown or not permitted for this User-Id

Response
Correct and try again.

POP0E16 `Incorrect authorization information`
Warranty: Y

Meaning
Either the User ID or password specified in the START-PROP-OBJECT-MONITORING command is incorrect

Response
Correct and try again

POP0001 `Subsystem ’(&00)’ version ’(&01)’ started`
Warranty: Y

POP0002 `Test message insert 0 ’(&00)’ insert 1 ’(&01)’ end`
Warranty: Y

POP0003 `Test message insert 3 '(&03)' insert 4 '(&04)' end`
Warranty: Y (C) Routing code: X Weight: 50

POP010A `Connection to administration entity already exists`
Warranty: Y

Meaning
The administration entity defined in the command was already defined and the connection is active.

POP010B `SDF communication area could not be set up correctly`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
IMPLEMENTOR=TPR(ENTRY=DUMMY,CALL=NEW) must be specified in the command definition.
A user command may not contain more than 20 operands.

Response
Correct the command and recompile with SDF-A.

POP010C `Syntax error in name`
Warranty: Y

Meaning
Possible cause: The REFERENCE-NAME specified cannot be abbreviated so that it is a valid reference name forUCON, or the name includes "$" or "-".

Response
Correct the name and retry the command.

POP0100 `Administration entity already in database`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
The administration entity definition in the command already exists.

Response
Check the name specified for the administration entity, definition and overwrite if required.

POP0102 `No administration entity definition`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
The administration entity definition in the command does not exist.

Response
Check the name specified for the administration entity, definition and correct it.

POP0107 `Administration entity not of type UCON`
Warranty: Y (B) Routing code: R Weight: 50

Meaning
Specified type is not supported in this PROP-XT version.

Response
Change the administration entity type.

POP0108 `No more administration processes/entities allowed`
Warranty: Y (B) Routing code: R Weight: 50

Meaning
The maximum number of administration processes/entities currently permissible have already signed on.

POP020F `Subsystem JV not available`
Warranty: Y

POP021D `Message code does not exist. Alternate message was generated`
Warranty: Y

Meaning
The message could not be found in the message files assigned.

Response
Define the message and/or attach the message file

POP0211 `Syntax error - Command privilege "PROP-ADMINISTRATION" does not exist`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
The command originator (administrator) does not have the "PROP-ADMINISTRATION" privilege.

Response
Assign the administrator the "PROP-ADMINISTRATION" privilege

POP0213 `Specified system message could not be generated`
Warranty: Y

Meaning
SEND-TO-PROP-ADMINISTRATOR
The specified system message could not be generated (Message with MSG-ID not in the message file or message file not assigned).

SIMULATE-PROP-EVENT
The message to be simulated could not be generated.
Error in accessing the message file. Check whether this file actually contains the message with the specified MSG-ID.

POP0214 `Syntax error - Command ignored`

Warranty: Y (C) Routing code: R Weight: 50

Meaning

The command syntax is incorrect. Detailed information on the syntax error is supplied in the subsequent messages.

Response

Correct and reenter the command.

POP022A `New databasefile was created`

(C) Routing code: R Weight: 50

POP0223 `Syntax error - SDF environment cannot be set up`

Warranty: Y (C) Routing code: R Weight: 50

Meaning

An error occurred while setting up the environment assigned to the command originator.

Response

Check the command originator’s user ID and set up correctly.

POP0225 `Access to job variable not possible`

Warranty: Y

Meaning

The job variable cannot be accessed (not defined, not shareable).

POP0229 `ENCRYPTION-PASSWORD (OLD-PASSWORD) specified for the required database file is incorrect`

(C) Routing code: R Weight: 50

Meaning

The passwords contained in the file were not all encrypted with the specified password, or the file is not a PROP-XT database file. The command was not executed.

POP0235 `Version of database file incorrect`

(C) Routing code: R Weight: 50

Meaning

PROP-XT cannot work with the version specified in the database file, or the file is not a PROP-XT database file. The command was not executed.

POP0240 `Specified timestamp ambigious`

Warranty: Y

Meaning

The time specified is in the time overlap when switching from daylight saving time to normal time. Depending on the SEASON operand, an alternative time has been fixed.

POP0241 `Calculated time ambiguous`
Warranty: Y

Meaning
The time specified is in the time overlap when switching from daylight saving time to normal time. Depending on the SEASON operand, an alternative time has been fixed.

POP0242 `Specified timestamp does not exist`
Warranty: Y

Meaning
Owing to the switch from normal to daylight saving time, the time specified for the timestamp does not exist. Depending on the SEASON operand, an alternative time has been fixed.

POP0705 `ENCRYPTION-PASSWORD (OLD-PASSWORD) password is different from the current one current one`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
The command was not executed. Changes to the database will only be implemented after the current password has been entered.

Response
Enter the correct password and try again.

POP0707 `Internal error when switching the database file`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Errors occurred in PROP-XT (DVS error or resource bottleneck
Possible reasons:
- The open DB file cannot be closed.
- The DB file cannot be created, opened, read or written to.
- The password cannot be encrypted.

Response
- Try again with a different database file.
- Notify the system administrator.

POP0708 `Internal error when accessing the database`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Errors occurred in PROP-XT (resource bottleneck, data inconsistency). The status of the database is undefined.

Response
Notify the system administrator.

`POP090B` `Internal error. Command not executed fully`

Warranty: Y (C) Routing code: R Weight: 50

Meaning

Errors occurred in PROP-XT (resource bottleneck, data inconsistency).
The command was not executed fully.

Response

Retry.

`POP090C` `No administration process specified in command exists`

Warranty: Y (C) Routing code: R Weight: 50

Meaning

None of the administration processes specified in the command is signed on to PROP-XT.

Response

Correct names of administration processes.

`POP090D` `Not all administration processes exist`

Warranty: Y (C) Routing code: R Weight: 50

Meaning

Not all the administration processes specified in the command are signed on to PROP-XT.

Response

Correct names of administration processes.

`POP090E` `No response administration entity definition`

Warranty: Y (C) Routing code: R Weight: 50

Meaning

No administration entity definition exists for the response administration entity specified in the command.

Response

Create an administration entity definition for the response administration entity.

`POP090F` `No administration entity definition for STD-ACTION administration entity`

Warranty: Y (C) Routing code: R Weight: 50

Meaning

No administration entity definition exists for the STD-ACTION administration entity.

Response

Create an administration entity definition for the STD-ACTION administration entity.

POP0900 `No such command`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
PROP-XT does not recognize the command entered.

Response
Correct the command if it is incorrect.

- Check whether PROP-XT can be administered. If it cannot, subsystem PROP-XT must be started again.

POP0901 `STD-ACTION already exists. Command not executed.`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
The STD-ACTION specified in the command is already signed on to PROP-XT.

Response
Check the name specified for STD-ACTION, and if required overwrite using the REPLACE-OLD-ACTION=YES operand.

POP0902 `Internal error. STD-ACTION could not be accepted`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Errors occurred in PROP-XT (resource bottleneck, data inconsistency).

POP0903 `STD-ACTION does not exist`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
The STD-ACTION specified in the command is not known to PROP-XT.

Response
Check and correct the name specified for the STD-ACTION.

POP0904 `Internal error. STD-ACTION could not be deleted`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Errors occurred in PROP-XT (resource bottleneck, data inconsistency).

POP0906 `None of specified STD-ACTIONs exists`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Specified STD-ACTIONs are not signed on to PROP-XT.

Response
Correct the names of the specified STD-ACTIONs.

POP0907 `Not all specified STD-ACTIONs exist`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Not all specified STD-ACTIONs are signed on to PROP-XT.

Response
Check the names of the STD-ACTIONs and retry command.

POP0908 `Internal error. Administration entity could not be accepted`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Errors occurred in PROP-XT (resource bottleneck, data inconsistency).

POP0909 `Internal error. Administration entity could not be deleted`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Errors occurred in PROP-XT (resource bottleneck, data inconsistency).

POP0910 `Specified type of STD-ACTION administration entity not supported`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Specified type is not supported in this PROP-XT version.

Response
Change the administration entity type.

POP0911 `No administration object specified in command exists`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
None of the administration objects specified in the command is signed on to PROP-XT.

Response
Correct names of administration objects.

POP0912 `Not all administration objects exist`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
Not all the administration objects specified in the command are signed on to PROP-XT.

Response
Correct names of administration objects.

POP0913 `No administration object specified`
(C) Routing code: R Weight: 50

POP9A04 `Event '(&01)' not known`
Warranty: Y

POP9A06 `Connection to administration entity '(&01)' not active`
Warranty: Y

POP9A07 `Administration entity '(&01)' not defined for administration process`
Warranty: Y

POP9F01 `Connection to administration entity '(&01)' could not be established`
Warranty: Y (C) Routing code: R Weight: 50

Meaning
No connection could be established to the administration entity specified in the STD-ACTION.

POP9F02 `Loss of connection to administration entity '(&01)' of STD-ACTION`
(C) Routing code: R Weight: 50

Meaning
Loss of data must be anticipated. PROP-XT initiates connection setup automatically when this is required.

POP9500 `PROP-XT cannot be administered`
(C) Routing code: R Weight: 50

POP990A `Error in command analysis for user '(&01)'`
(C) Routing code: R Weight: 50

Meaning
Check whether the command is contained in the relevant SDF syntax file. As command analysis of PROP-XT user commands is performed in both the command originator's environment and in the command-processing administration procedure, you are notified here of the environment in which errors occurred.

POP990C `Administration object '(&01)' not known`
(C) Routing code: R Weight: 50

Meaning
Specified object is not signed on to PROP-XT.

Response
Correct the object name in the command.

POP9900 `All PROP-XT administration commands available`
(C) Routing code: R Weight: 50

Meaning
PROP-XT can now be administered.

POP9902 `STD–ACTION '(&01)' not known`
(C) Routing code: R Weight: 50

Meaning
STD-ACTION is not signed on to PROP-XT.

Response
Sign on STD-ACTION or correct name of STD-ACTION in command.

POP9903 `None of specified processes known`
(C) Routing code: R Weight: 50

Meaning
None of the specified processes is signed on to PROP-XT.

Response
Correct the process name in the command.

POP9904 `Administration process '(&01)' not known`
(C) Routing code: R Weight: 50

Meaning
Specified process is not signed on to PROP-XT.

Response
Correct the process name in the command.

POP9905 `No administration entity exists for administration process '(&01)'`
(C) Routing code: R Weight: 50

Meaning
The specified administration process is not monitoring any administration entities.

POP9906 `No event exists for administration process '(&01)'`
(C) Routing code: R Weight: 50

Meaning
Specified administration process is not monitoring any events.

POP9908 `Administration process '(&01)' could not be notified`
(C) Routing code: R Weight: 50

Meaning
The specified administration process could not be notified by setting a return code.

POP9909 `Administration process '(&01)' could be notified`
(C) Routing code: R Weight: 50

Meaning
The specified administration process was notified by setting a return code.

Glossary

administration commands

Operator special commands used to control PROP-XT, obtain information, and define and manage standard tasks. Administration commands are processed by PROP-XT itself.

administration entity

Object managed by means of intervention by *administrators*.

administration procedure

BS2000 procedure containing PROP-XT *system commands*.

administration process

BS2000 user process in which an *administration procedure* runs.

administrators

With PROP-XT, all entities which manage *administration entities*, especially PROP-XT, and *administration processes*. Administrators can therefore be both persons using the system, and authorized user processes.

events

An *administration process* is activated when one of a set of predefined events occurs. An *administration process* must define which events cause it to be activated. PROP-XT distinguishes between the following event types:
- message events
- command events
- job variable events
- time events
- job events

job types

For *administration entities* of the type OPERATING, PROP-XT supports the job types defined by UCON:
- messages (individual notifications)
- questions (messages that cause a response)
- commands (multi-level message sequences)

jobs

Jobs are special *messages* and message sequences. Only *administration entities* of the type OPERATING are affected.

messages

Messages that are exchanged between *administrators* and *administration procedures*, or between *administration procedures* and *administration entities*, can have no type, or can be of a *job type* supported by UCON.

PROP-XT commands

All commands processed by PROP-XT.
A distinction is made between *system commands*, *administration commands*, and *user-defined commands*.

system commands

Commands of the PROP-XT programming interface, which can be used to create *administration procedures*.

system operator

Person who, amongst other things, provides the generation data needed to run PROP-XT in a database, monitors PROP-XT in actual operation and, if required, intervenes to control PROP-XT using *administration commands*.

user-defined commands

Operator special commands defined by programmers themselves using SDF-A. User-defined commands sign on to PROP-XT by means of *events*. They are entered by *administrators*, and are transferred to the *administration procedure* at a defined wait point.

Related publications

AVAS (BS2000)
Job Management and Handling System
Introduction

Target group
Anyone wanting to get acquainted with AVAS
Contents
This manual provides an introduction to the AVAS Job Management and Handling System. It describes the individual work steps involved, outlines the tasks of AVAS management and provides information on system security. The AVAS user guide can be used as a reference work.

BS2000/OSD-BC V2.0A
Commands, Volume 1, A-L
User Guide

Target group
The manual addresses both nonprivileged users and systems support.
Contents
This manual contains the commands ADD-... to LOGOFF (BS2000/OSD basic configuration and selected products, with the functionality for all privileges). The introduction provides information on command input.

BS2000/OSD-BC V2.0
Commands, Volume 2, M-SG
User Guide

Target group
The manual addresses both nonprivileged users and systems support.
Contents
This manual contains the commands MODIFY... to SET... (BS2000/OSD basic configuration and selected products, with the functionality for all privileges).

BS2000/OSD-BC V2.0A
Commands, Volume 3, SH-Z
User Guide

Target group
The manual addresses both nonprivileged users and systems support.
Contents
This manual contains the commands SHOW-... to WRITE-... (BS2000/OSD basic configuration and selected products), with the functionality for all privileges. By means of SDF-P users of SHOW commands can make use of output in structured S variables which are described in Volume 4.

BS2000/OSD-BC
System Operator's Guide
User Guide

Target group
This manual addresses operators at installations of the BS2000/OSD operating system.
Contents
It describes the operator's functions and responsibilities as well as the commands available for this purpose at the operator terminal. The following items are dealt with:

- System initialization and termination (startup types, shutdown)
- Commands in alphabetical order
- Device management (reconfiguration, resource allocation, volume monitoring, NDM handling, duplex reconfiguration)
- Tools and methods for facilitating system operation
- Memory dumps (SLED)
- Messages and responses in the event of saturation states

BS2000/OSD-BC V2.0A
Subsystem Management (DSSM/SSCM)
User Guide

Target group
The manual addresses systems support and software consultants of BS2000.
Contents
The following are described: subsystem concept of BS2000, dynamic subsystem management DSSM V3.5, subsystem catalog management SSCM V2.0 and the associated commands and statements.

BS2000/OSD-BC V2.0A
Introductory Guide to Systems Support
User Guide

Target group
The manual addresses BS2000/OSD systems support and operators.
Contents
The manual contains the following topics on management and monitoring of the BS2000/ OSD basic configuration: system introduction, parameter service, job and task control, memory/device/user/file management, assignment of privileges, accounting and operator functions.

JV V11.2A (BS2000/OSD)
Job Variables
User Guide

Target group
The manual addresses both nonprivileged users and systems support.
Contents
The manual describes the management and possible uses of job variables. The command descriptions are divided according to function areasof the job variables. The macro calls are described in a separate chapter.

OMNIS (TRANSDATA, BS2000/OSD)
Functions and Commands
User Guide

Target group
- OMNIS administrators
- OMNIS users

Contents
- Essential information for the use of OMNIS
- Description of the functions provided and possible implementations of OMNIS

OMNIS (TRANSDATA, BS2000)
Administration and Programming
User Guide

Target group

- OMNIS administrators
- Programmers

Contents
Introduction to OMNIS administration, the OMNIS utility routines and the application interface for extending the OMNIS functionality

OMNIS-PROP (TRANSDATA, BS2000)
User Guide

Target group
System administrators, system operators and persons entrusted with such tasks.
Contents
OMNIS-PROP is used for automating system operation in a computer center. It is based on OMNIS and provides a programming aid for routines for controlling and administering programmed operator measures, as well as ready-made and simple standard solutions for areas of system operation.

SDF-A (BS2000/OSD)
User Guide

Target group
The manual is intended for experienced BS2000 users and system administration staff.
Contents
It describes how to process syntax files and explains the SDF-A functions on the basis of examples. All SDF-A statements are listed in alphabetical order. SDF-A V4.0A can be used in BS2000/OSD-BC V1.0 and V2.0.

SDF-P V2.0A (BS2000/OSD)
Programming in the Command Language
User Guide

Target group
The manual addresses BS2000/OSD users and systems support.
Contents
SDF-P is a structured procedure language in BS2000. The introduction is followed by a detailed description of commands, functions and macros. SDF-P V2.0A can be used under BS2000/OSD-BC V1.0 only with VAS 2.0A and SDF V4.0.

SECOS (BS2000/OSD)
Security Control System
User Guide

Target group

- BS2000 system administrators
- BS2000 users working with extended access protection for files

Contents

Capabilities and application of the four functional units:

- SRPM (System Resources and Privileges Management)
- FACS (File Access Control System)
- GUARDS (Generally Usable Access Control Administration System)
- SAT (Security Audit Trail).

Ordering manuals

The manuals listed above and the corresponding order numbers can be found in the Siemens Nixdorf *List of Publications*. New publications are described in the *Druckschriften-Neuerscheinungen (New Publications)*.

You can arrange to have both of these sent to you regularly by having your name placed on the appropriate mailing list. Please apply to your local office, where you can also order the manuals.

Index

Contents

PROP-XT V1.1A (BS2000/OSD)

Programmed Operating with SDF-P

Target group
System administrators, system users, work process schedulers, and anyone involved in similar activities.
Contents
PROP-XT permits programmed operating by the user by means of administration procedures created with the user-friendly language elements of SDF-P. It also allows simple administrative activities to be automated using special administration commands.

Edition: July 1995

File: PROP_XT.PDF

Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format *…@ts.fujitsu.com*.

The Internet pages of Fujitsu Technology Solutions are available at
http://ts.fujitsu.com/...
and the user documentation at *http://manuals.ts.fujitsu.com*.

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form *…@ts.fujitsu.com*.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter
http://de.ts.fujitsu.com/..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.