
ESA/TOS-GCI

SCOS-2000
Database Import ICD



Document Reference:

S2K-MCS-ICD-0001-TOS-GCI

Document Status:

Issue 5.1

Prepared By:

SCOS-2000 Team

Date:

October 26, 2001

	Name	Date	Signature
Prepared by :	SCOS -2000 Team		
Verified by :	M. Schick, TERMA GmbH		
Approved by :	S. Haag, TOS-GCI		
Authorised by :	M. Pecchioli, TOS-OFM M. Jones, TOS-GCI		

Distribution List

TOS-GCI	ESA/TOS-GCI SCOS -2000 Industry Team SCOS -2000 Document Library
---------	--

© COPYRIGHT EUROPEAN SPACE AGENCY, 1998

The copyright of this document is vested in European Space Agency. This document may only be reproduced in whole or in part, stored in a retrieval system, transmitted in any form, or by any means electronic, mechanical, photocopying, or otherwise, with the prior permission of the owners.

DOCUMENT CHANGE LOG

Issue/ Revision	Date	Modification Nb	Modified pages	Observations
3.2	12/03/99		All	First official issue for SCOS-2000.
3.3	09/07/99		All	Conversion into Microsoft Word (internal distribution only)
4.0	29/10/99		All	Many editorial changes plus some non-editorial changes as listed in Annex 1 of this document.
4.1	20/12/99		See change bars	Updates required to clarify the actual implementation. See Annex 1 for a complete list of implemented change requests.
4.2	01/12/00		See change bars	Updates required to clarify the actual implementation. Modifications describing the changes introduced in R2.2. See Annex 1 for a complete list of implemented DCRs.
5.0	02/04/01		See change bars	Updates required to clarify the actual implementation. Modifications describing the changes introduced in SCOS-2000 R.2.2e (mainly related to the addition of the SCOS-2000 Packetiser). See Annex 1 for a complete list of implemented Document Changes.
5.1	26/10/01		See change bars	Updates required to clarify the actual implementation. Modifications describing the changes introduced in SCOS-2000 R.2.3 (which will be implemented in R3.0) as well as the specific additions implemented in the evolution version for EGSE systems.



PAGE ISSUE RECORD

Issue of this document comprises the following pages at the issue shown

Page	Issue/ Revision	Page	Issue/ Revision	Page	Issue/ Revision	Page	Issue/ Revision	Page	Issue/ Revision



Abstract

This document is the interface control document of the data import into SCOS-2000 run-time database. Data in the suitable format are typically generated by the off-line database management system of any client mission using SCOS-2000. This document only covers the 'generic' portion of the static database which are relevant to SCOS-2000 functions. Mission specific extensions are possible.



TABLE OF CONTENTS

ABSTRACT..... V

LIST OF TABLES VIII

1 INTRODUCTION1

1.1 PURPOSE1

1.2 SCOPE.....1

1.3 OVERVIEW.....1

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....2

1.4.1 *Definitions and Terminology*.....2

1.4.2 *Acronyms and Abbreviations*.....2

1.5 REFERENCES.....2

1.5.1 *Applicable documents*2

1.5.2 *Reference documents*3

2. INTERFACE OVERVIEW4

2.1 ASSUMPTIONS AND CONSTRAINTS.....4

2.2 LOCATION AND NAMING CONVENTION OF THE ASCII FILES4

2.3 STRUCTURE OF THE ASCII FILES4

2.4 CHECKS APPLIED AT IMPORT TIME.....5

2.5 IMPORT OPTIONS.....5

3. DETAILED INTERFACE SPECIFICATION6

3.1 RELATIONSHIP BETWEEN THE ASCII TABLES AND THE DATABASE ITEMS 6

3.2 SUMMARY OF THE ASCII TABLES.....9

3.3 DETAILED DEFINITION OF THE ASCII TABLES STRUCTURE12

3.3.1 *General*.....14

3.3.1.1 Database version: vdf.....14

3.3.2 *Monitoring Data*.....14

3.3.2.1 Monitoring Parameters.....15

3.3.2.1.1 Monitoring parameters characteristics: pcf.....15

3.3.2.1.2 Synthetic parameters20

3.3.2.1.3 User Defined Constants.....21

3.3.2.1.4 Dynamic Configuration Variables.....23

3.3.2.2 Monitoring parameters calibration24

3.3.2.2.1 Numerical calibration curves: caf.....24

3.3.2.2.2 Numerical calibration curves definition: cap25

3.3.2.2.3 Textual calibrations: txf.....25

3.3.2.2.4 Textual calibrations definition: txp26

3.3.2.2.5 Polynomial calibration curves: mcf.....26

3.3.2.3 Monitoring parameter checks.....27

3.3.2.3.1 Monitoring checks: ocf.....27

3.3.2.3.2 Monitoring checks definition: ocp.....28

3.3.2.4 Telemetry Packets Definition.....30

3.3.2.4.1 Packets identification: pid30

3.3.2.4.2 Packets identification criteria: pic34

3.3.2.4.3 Telemetry packets characteristics: tpcf.....35

3.3.2.5 Telemetry Packets Processing.....36

3.3.2.5.1 Parameters location in fixed packets: plf36

3.3.2.5.2 Variable packet definition: vpd.....37

3.3.2.6 Displays and reports42

3.3.2.6.1 Alphanumeric displays: dpf.....42

3.3.2.6.2 Alphanumeric displays definition: dpc43



- 3.3.2.6.3 Graphic displays: gpf.....44
- 3.3.2.6.4 Graphic displays definition: gpc.....46
- 3.3.2.6.5 Scrolling displays: spf.....47
- 3.3.2.6.6 Scrolling displays definition: spc48
- 3.3.2.6.7 Printout proforma: ppf.....49
- 3.3.2.6.8 Printout proforma definition: ppc.....50
- 3.3.3 Commanding data.....51
 - 3.3.3.1 Packet Headers52
 - 3.3.3.1.1 Packet headers characteristics: tcp52
 - 3.3.3.1.2 Packet headers parameters: pcpc.....52
 - 3.3.3.1.3 Packet headers definition: pcpf53
 - 3.3.3.2 Commands.....54
 - 3.3.3.2.1 Command characteristics: ccf.....54
 - 3.3.3.2.2 Command parameters: cpc.....59
 - 3.3.3.2.3 Commands definition: cdf.....62
 - 3.3.3.2.4 Commands pre -transmission validation: ptv65
 - 3.3.3.3 Command Sequences66
 - 3.3.3.3.1 Command sequence characteristics: csf.....69
 - 3.3.3.3.2 Command sequences definition: css.....70
 - 3.3.3.3.3 Command sequence element parameters: sdf.....79
 - 3.3.3.3.4 Command sequence formal parameters: csp.....81
 - 3.3.3.4 Verification.....85
 - 3.3.3.4.1 Verification stages file: cvs.....85
 - 3.3.3.4.2 Verification expressions: cve86
 - 3.3.3.4.3 Verification profiles: cvp88
 - 3.3.3.5 Command/Sequence Parameter Sets.....89
 - 3.3.3.5.1 Parameter sets: pst.....89
 - 3.3.3.5.2 Parameter value sets: psv89
 - 3.3.3.5.3 Parameter sets definition: cps.....89
 - 3.3.3.5.4 Parameter value sets definition: pvs.....90
 - 3.3.3.5.5 Parameter sets mapping to commands/sequences: psm.....91
 - 3.3.3.6 Command/Sequence Parameters (De-)calibrations91
 - 3.3.3.6.1 Numerical (de-)calibration curves: cca.....91
 - 3.3.3.6.2 Numerical (de-)calibration curves definition: ccs92
 - 3.3.3.6.3 Textual (de -)calibrations: paf.....93
 - 3.3.3.6.4 Textual (de -)calibrations definitions: pas.....93
 - 3.3.3.7 Command/Sequence Parameters Range Checks94
 - 3.3.3.7.1 Parameter range sets: prf.....94
 - 3.3.3.7.2 Parameter range values: prv95
- APPENDIX A – SCOS-2000 PARAMETER TYPES 97**
- APPENDIX B– SCOS-2000 DATABASE IMPORT TABLES OVERVIEW105**
- ANNEX 1 – LIST OF IMPLEMENTED DOCUMENT CHANGES110**
- ANNEX 1 – LIST OF IMPLEMENTED DOCUMENT CHANGES111**
- MONITORING.....111**



List of Tables

Table 1 - Summary of Monitoring related data.....	7
Table 2 - Summary of Displays related data	7
Table 3 - Summary of Commanding related data.....	8
Table 4 - Parameter Type and Format Codes supported by SCOS-2000	102
Table 5 - Value formats for the SCOS-2000 parameter types.....	104

1 INTRODUCTION

1.1 PURPOSE

This document is the interface control document of the data import into the SCOS-2000 run-time database. Data in the suitable format are typically generated by the off-line database management system of client missions using SCOS-2000.

1.2 SCOPE

This document describes how a client mission using the SCOS-2000 system (later referred as client mission) has to present its data so that it can be imported by the SCOS-2000 system (referred as SCOS-2000 in the document). This document only covers the 'generic' portion of the static database which are relevant to SCOS-2000 functions. Mission specific extensions are possible.

The SCOS-2000 client mission MIB static data are typically imported/activated as follows:

- the client mission MIB static data are converted/exported from the off-line database system (based on e.g. ORACLE, MS-Access, etc.) into ASCII files conforming to the structure specified in this document
- these resulting ASCII files are transferred into a specific directory accessible by the SCOS-2000 importer.
- the ASCII files are then imported into the SCOS-2000 run-time database format and location.
- the imported data are used for processing as soon any application requiring them is restarted.

This document only addresses the definition of the ASCII files to be imported into the SCOS-2000 run-time database format.

1.3 OVERVIEW

This document is split into:

- Introduction: this section
- Interface overview
- Detailed interface specification (i.e. description of the Monitoring and Commanding Tables)
- Appendix A specifying the parameter types supported by SCOS-2000
- Appendix B providing a graphical overview of all tables and of the relationships between them.

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

1.4.1 Definitions and Terminology

The SCOS-2000 Mission Information Base is split into a static part (typically referred to as the 'database') and a dynamic part (typically referred to as the 'history files').

Sometimes the term "ASCII file" will be used instead of ASCII table. This is because each ASCII file consists of a single table.

The term 'raw value' is used to indicate a parameter value prior to calibration (for monitoring parameters) or resulting from a de-calibration (for command/sequence parameters). The term 'engineering value' is used to indicate the parameter calibrated value. For synthetic parameters in particular, the raw value indicates the output of the associated (OL or hard-coded) routine.

Remark: The raw value should not be confused with the 'encoded' value i.e. the actual bits which get down-linked/up-linked in a TM or TC packet. The raw value and the encoded value only coincide for parameters of type unsigned integer. For other parameter types the encoded value gets automatically 'decoded/ encoded' by SCOS-2000 applications according to the parameter type (i.e. signed integer, real, octet string, character string, time and delta time). For example, the encoded value corresponding to a raw value '-1' of a 16-bit signed integer parameter is FFFF Hex. The encoded values are not available for display.

The term 'name' is normally used to indicate the unique ID of an item (e.g. a parameter) if this is expressed as an alphanumeric string (typically 8 characters long). In case the unique ID of an item is a number (e.g. a calibration curve), the term 'identifier' will instead be used.

A complete definition of the specific terms used within the SCOS -2000 Project is available in [RD-3].

1.4.2 Acronyms and Abbreviations

A complete list of the Acronyms and Abbreviations used within the SCOS-2000 Project is available in [RD-3]. In this document, the acronym of the ASCII tables (e.g. pcf) are specified in lower or upper case without any special rule. However, the name of the files to be imported have to be in lower case as the operating system is case sensitive (see Section 2.2 below).

1.5 REFERENCES

1.5.1 Applicable documents

Doc.	Reference	Title
AD-1	S2K-MCS-SRD-0003-TOS-GCI	SCOS-2000 Monitoring SRD
AD-2	S2K-MCS-SRD-0002-TOS-GCI	SCOS-2000 Commanding SRD
AD-3	S2K-MCS-SRD-0001-TOS-GCI	SCOS-2000 System Level SRD

1.5.2 Reference documents

Doc.	Reference	Title
RD-1	S2K-MCS-SUM-0002-TOS-GCI	SCOS-2000 Configuration and Installation Guide
RD-2	S2K-MCS-ADD-0007-TOS-GCI	SCOS-2000 Operations Language ADD
RD-3	S2K-MCS-GLO-0001-TOS-GCI	SCOS-2000 Glossary, Definitions and Acronyms
RD-4	TERMA/SSSL/IMCS/MAN/0002	Integral Mission Control System Database Subsystem SUM, Issue 1.0
RD-5	Deleted	
RD-6	ESA PSS-07-101	Packet Utilisation Standard, Issue 1, May 1994
RD-7	RO-ESC-IF-5002	Rosetta Space/Ground ICD, Issue 1, August 1998
RD-8	CCSDS 301.0-B-2	CCSDS Time Code Formats, Issue 2, April 1990
RD-9	S2K-MCS-SUM-0019-TOS-GCI	SCOS-2000 Synthetic Parameters SUM
RD-10	S2K-MCS-SUM-0026-TOS-GCI	SCOS-2000 Events & Actions SUM
RD-11	S2K-MCS-SUM-0018-TOS-GCI	SCOS-2000 MIB Applications SUM
RD-12	S2K-MCS-ICD-0002-TOS-GCI	SCOS-2000 Stack Import ICD
RD-13	S2K-MCS-ICD-0003-TOS-GCI	SCOS-2000 Task Parameter File ICD
RD-14	ESA PSS-074-107	ESA Packet Telecommand Standard, Issue 2.0, April 1992
RD-15	S2K-MCS-SUM-2210-TOS-GCI	SCOS-2000 Telemetry Desktop Operator User Manual

2. INTERFACE OVERVIEW

The data source consists of a set of ASCII files describing the static part of the SCOS-2000 Mission Information Base. One ASCII file will contain all data belonging to one table. Definitions of Dynamic OL Synthetic Parameters are also stored in dedicated files (one file per Synthetic Parameter).

These files are interpreted by the SCOS-2000 system to build the run-time SCOS -2000 database.

2.1 ASSUMPTIONS AND CONSTRAINTS

The ASCII files will have to respect the exact structure described in the present document. Each ASCII file provided by the mission consists of a set of records all based on the same structure.

Some of the ASCII files detailed within this ICD require to be sorted, since some of the tables do not contain position fields. The off-line database system of client missions are supposed to deliver the ASCII files with the contained records appropriately sorted (this is only needed if explicitly specified so in this ICD for each ASCII table).

This ICD does not impose any specific constraint in the way that the ASCII files are generated and maintained in the off-line database system. Typically, a relational database system will be used for this purpose (e.g. ORACLE, MS ACCESS).

2.2 LOCATION AND NAMING CONVENTION OF THE ASCII FILES

The transfer of the database ASCII files will have to be performed by the mission onto its own SCOS-2000 system. The name and path of the directories where the ASCII files have to be located is configurable by client missions.

One directory will contain all ASCII files corresponding to the database tables specified in the next section (typically /home/sops1/data/ASCII). A dedicated sub-directory (typically /home/sops1/data/ASCII/SYNTHETIC) will contain the synthetic parameter files (each containing the Operations Language expression for Dynamic OL Parameters, see section 3.3.2.1.2 below).

Each ASCII file shall be given the name of the corresponding table in lower case with an extension .dat (e.g. pcf.dat). The files containing the OL expressions of dynamic OL Synthetic Parameters shall be given the name of the corresponding parameter (PCF_NAME) without any file extension.

2.3 STRUCTURE OF THE ASCII FILES

Each ASCII file corresponding to the database tables specified in the next section will contain one line per table record.

The field separator is a Tabulation.

Remark: a client mission can override this and use a different separator. This separator has to be then adjusted within the SCOS-2000 import program.

Fields that are required by client missions and are not part of the tables described below, have to be added at the end of the table.

The maximum length of each field imported by SCOS-2000 is specified in Section 3.3 below. The maximum allowed length for mission specific fields is only restricted by the maximum string size supported by the computer where the importer software runs.

The end-of-file convention is EOF.

2.4 CHECKS APPLIED AT IMPORT TIME

This section describes the checks which are performed at import time by the SCOS-2000 importer. These checks are only meant to spot the most dangerous inconsistencies in the database ASCII files which certainly lead to processing problems. However, the importer has not been designed to identify all possible inconsistencies. **It is the responsibility of the off-line database system to ensure consistency of the database tables to be imported into SCOS-2000.** An example of database consistency checks is available in [RD-4].

Every imported ASCII file is checked against the field characteristics in the corresponding table specified in this document. Anomalies in the data files are reported with warning message during the import phase, provided that the appropriate flag is specified when launching the import process. All messages generated by the importer are logged in an ASCII file which is overwritten at each import operation (/home/sops1/Import.log).

If an imported field exceeds the allocated length it is truncated and a warning message is generated. The truncation may have further effects during processing (unless the truncated field is a textual description which is used for display purposes only).

If an imported field which is flagged as mandatory in this ICD (i.e. marked with an "M" in the 'M/Def' column, Chapter 3 below) is left null, a warning message is generated and the importer sets the corresponding value in SCOS-2000 to '0' (zero) for numbers and to "" (null) for strings. This should never happen in order to avoid further failures during processing.

During the import process, relational integrity between tables is also checked. If a record creates inconsistency (e.g. it refers to a parameter that is not present in the database) it is not imported and a warning message is generated.

All values for command and sequence parameters are also syntactically checked by the importer against the applicable format (see Appendix A). In case the value is not compatible with the required format (e.g. integer, real, time), a warning message is generated and the parameter value is set to '0' (zero) for integer, real and delta times, to "" (null) for ASCII strings and to '1970.001.00.00.00' for absolute times.

2.5 IMPORT OPTIONS

The SCOS-2000 database importer supports the following options (see [RD-11] for a complete description of the SCOS-2000 Database Import software):

- Full database import: this option creates a completely new internal SCOS-2000 database (default)
- Monitoring Tables only: this option leaves the current internal content of commanding related data unaffected. This option includes import of the monitoring displays
- Commanding Tables only: this option leaves the current internal content of monitoring related data unaffected
- Selected Tables only: this option is only available for a subset of the imported tables (see [RD-11])
- 'Check only': this option does not affect any data but just reports about the outputs of the tables structure and field length checks performed by the importer.

3. DETAILED INTERFACE SPECIFICATION

This chapter provides a detailed description of the ASCII tables which will be imported into SCOS-2000 run-time database.

3.1 RELATIONSHIP BETWEEN THE ASCII TABLES AND THE DATABASE ITEMS

The following tables list the items (e.g. parameters, displays, commands, sequences) whose characteristics are defined via the static database and their relationship with the ASCII Tables which are imported into SCOS-2000.

When 'all' is stated in the following tables, it means that all the fields contribute to define the database item e.g. all the fields of the PCF table are needed to fully describe a parameter definition.

Remark: all tables have been defined using a 'normalised' structure i.e. each field contains a single entity (and never a variable list of them). Many-to-many relationships between tables are avoided by means of 'mapping' tables.

MIB Item	ASCII table	Field	Comments
Parameters Definition	pcf	All	
Synthetic parameters			
	Definition	pcf	All
	OL expression	File whose name is the Parameter name	OL expression Only for Dynamic OL Synthetic Parameters (i.e. PCF_NATUR = 'D').
User Defined Constants			
	Definition	pcf	All
	Saving	plf	All UDCs are stored using standard TM packets.
Parameters Validity	pcf	PCF_VALID PCF_VALPAR	The validity parameter name and value.
Parameters Calibration			
	Numerical	caf, cap	All
	Textual	txf, txp	All
	Polynomial	mcf	
Applicability criteria			
	For OOL	ocp	OCP_RLCHK OCP_VALPAR Applicability parameter and value for the set of limits.
Parameters Checking			
	Limits	ocf, ocp	OCF_NBOOL OCF_INTER OCP_LVALU OCP_HVALU
	Status consistency	pcf, ocp	PCF_USCON OCP_TYPE
Packets definition	pid, tpcf	All	
Parameters extraction	plf, vpd	All, subset	The plf table is used to extract parameter samples out of TM packets with a fixed structure. The vpd table is used to extract parameter samples out of TM packets with a variable structure.

Table 1 - Summary of Monitoring related data

MIB Item	ASCII Table	Field	Comments
Database Version Definition	vdf	VDF_NAME	
Variable Packet definition	vpd	Subset	The vpd contains both information about how to extract parameters from variable TM packets and how to display them.
Alphanumeric Display definition	dpf, dpc	All	The dpf, dpc is not used to define the contents of the TM query display, neither detailed OOL display.
Graphics Display definition	gpf, gpc	All	
Scrolling Display definition	spf, spc	All	

Table 2 - Summary of Displays related data

MIB Item	ASCII Table	Field	Comments
Packet header			
Packet header definition	tcp	All	
Packet header parameters	pccp	All	
Packet header structure	pcdf	All	
Command			
Command definition and routing	ccf, dst	All	
Command parameters	cpc	All	
Command structure	cdf	All	
PTV checks	ptv	All	
Command Sequences			
High level sequence definition	csf	All	
Sequence elements definition	css	All	
Values of the elements parameters	sdf	All	
Sequences (formal) parameters	csp	All	
Verification			
CEV stages definition	cvs	All	
Verification expressions	cve	All	For monitoring parameters based checks only.
Verification profiles	cvp	All	
Parameter Sets			
Parameter sets definition	pst	All	
Parameter value sets definition	psv	All	
Parameter sets details	cps	All	
Parameter value sets details	pvs	All	
Mapping of parameter sets to commands/sequences	psm	All	
Parameters (De-)calibration			
Numerical curve definition	cca	All	
Numerical curve values	ccs	All	
Text curve definition	paf	All	
Text curve values	pas	All	
Parameters Range Checks			
Range sets definition	prf	All	
Range values	prv	All	

Table 3 - Summary of Commanding related data

3.2 SUMMARY OF THE ASCII TABLES

This section lists the ASCII tables which are imported into SCOS-2000. A detailed definition of each table is provided in the next section.

➤ General

- vdf: Database Version Definition file, containing the identifier and description of the database version. See page [14](#).

➤ Monitoring

- pcf: parameter characteristics file, containing the definition of monitoring parameters. See page [15](#).
- caf: calibration curve file, defining the numerical calibration curves. See page [24](#).
- cap: calibration curve definition file, defining all the raw/engineering value couples for each numerical calibration curves. See page [25](#).
- txf: text strings calibration curve file, defining the textual calibration curves. See page [25](#).
- txp: text strings calibration curve definition file, defining all the raw/string value couples for each textual calibration curves. See page [25](#).
- mcf: polynomial calibration curve definitions, defining the coefficients of the polynomial function used for calibration. See page [26](#).
- ocf: OutOfLimits checks file, defining the characteristics of all the checks applied to a specified monitoring parameter. See page [27](#).
- ocp: OutOfLimits definition file, defining the allowed (ranges of) values for monitoring parameters. See page [28](#).
- pid: packet identification file, containing the definition of TM packets and their correspondence with the packet identification fields (e.g. APID/type/subtype). See page [30](#).
- pic: packet identification criteria file, containing the definition and position of the additional identification fields for each packet type/subtype combination. See page [34](#).
- tpcf: telemetry packets characteristics file, defining the attributes of the SCOS-2000 Telemetry Packets. See page [35](#).
- plf: parameter location file, defining the location of the parameters in the fixed TM packets. See page [36](#).
- vpd: variable packet definition file, detailing the contents of variable TM packets. See page [37](#).

➤ Displays

- dpf: alphanumeric display proforma file, containing the list of TM alphanumeric displays (AND). See page [42](#).
- dpc: alphanumeric display proforma definition file, containing the list of parameters to be displayed in each AND. See page [43](#).
- gpf: graphic display proforma file, containing the list of TM graphic displays (GRD). See page [44](#).
- gpc: graphic display proforma definition file, containing the list of parameters to be displayed in each GRD. See page [46](#).
- spf: scrolling display proforma file, containing the list of TM scrolling displays (SCD). See page [47](#).

- spc: scrolling display proforma definition file, containing the list of parameters to be displayed in each SCD. See page 48.

➤ Commanding

- tcp: packet header file which defines TC packet headers. See page 52.
- pcpc: packet header parameter characteristics file which defines the TC packet header parameters. See page 52.
- pcdh: packet headers definition file which defines the structure of each packet header. See page 53.
- ccf: command characteristics file which defines the commands. See page 54.
- **dst: [command routing table which defines the destination of the commands. See page 58.](#)**
- cpc: command parameter characteristics which defines the editable command parameters. See page 59.
- cdf: command details file which defines the structure of the command application data field. See page 62.
- ptv: command pre-transmission validation file which defines the monitoring parameter and value pairs to satisfy validation. See page 65.
- csf: command sequence file which defines the command sequences. See page 69.
- css: command sequence set which defines the elements (commands or sequences) used in a command sequence. See page 70.
- sdf: sequence details file which defines the values for the editable parameters of all elements contained in a command sequence. See page 79.
- csp: command sequence parameter file which defines the command sequence (formal) parameter. See page 81.
- cvs: verification stages file which defines the verification stage details. See page 85.
- cve: verification expression file which defines the monitoring parameter and value pairs to satisfy verification. See page 86.
- cvp: command/sequence verification profiles file which defines the mapping of verification stages with commands/sequences. See page 88.
- pst: command/sequence parameter set file which defines the parameter sets characteristics. See page 89.
- psv: command/sequence parameter value set file which defines the parameter value sets characteristics. See page 89.
- cps: command/sequence parameter set file which defines the parameters contained in a parameter set. See page 89.
- pvs: command/sequence parameter value set file which defines the parameter forming a parameter value set. See page 90.
- psm: parameter sets mapping file which defines the mapping between parameter sets and tasks (i.e. commands or sequences). See page 91.
- cca: parameter calibration curve file which defines the numerical (de-)calibration (for command or sequence parameters). See page 91.
- ccs: calibration curve set file which defines the numerical (de-)calibration values. See page 92.



- paf: parameter alias file which defines the text (de-)calibration (for command or sequence parameters). See page [93](#).
- pas: parameter alias set which defines the text (de-)calibration values. See page [93](#).
- prf: parameter range file which defines the command/sequence parameter range checks. See page [94](#).
- prv: parameter range value file which defines the parameter allowed value ranges. See page [95](#).

3.3 DETAILED DEFINITION OF THE ASCII TABLES STRUCTURE

In the following sections, the structure of the records for each ASCII table supported by the SCOS-2000 database importer is described in a tabular form. The tables have to be read as follows:

- Each field of an ASCII table record is given a position, a name, corresponding type and description, and an indication of whether the field is mandatory (i.e. it cannot be left Null).
- Fields whose position number is followed by a '*' have to be considered as table keys i.e. the table is only allowed to contain one single record with a particular combination of values for the field keys (e.g. as the plf table is keyed on PLF_NAME and PLF_SPID, it is only allowed to contain one entry of a particular telemetry parameter in a particular SCOS-2000 Packet).
- The field type is specified to be either 'Number' or 'Char'. Only integer values (signed or unsigned) are allowed to be specified for field of type Number. Field of type Char allow any alphanumeric ASCII characters i.e. A-Z and 0-9. Underscore, plus/minus, dot and dash characters are also allowed. Other ASCII characters shall be avoided. Also note that the format of a specific field can be further constrained by its nature e.g. parameter values shall be expressed in the specific format corresponding to the parameter type and value representation (as specified in Appendix A).

Remark: It is recommended not to use 'space' characters within names i.e. within the alphanumeric strings used for unique identification of a database item (e.g. a parameter name). Further, the uniqueness of names shall be ensured without considering the case of alphabetical characters e.g. two parameters with names A1000 and a1000 are not allowed. No warning message is generated by the importer in case two items are given names which only differ in the case used for alphabetical characters.

- All the field lengths have to be considered as a maximum length for that field i.e. it is allowed to specify values with less digits/characters than the maximum field length. **In case the actual allowed range for the field value is more restrictive than the declared field length, this is explicitly stated as part of the field description. If no explicit range is specified as part of the field description, any value compatible with the applicable format and fitting within the maximum field length is allowed.**
- If a row is with grey background, it means that the field will not be used by SCOS-2000 for any processing. Typically, these fields are required by the off-line database system to perform consistency checking or have been left in the tables definition for historical reasons.
- If some text within the tables definition or an entire row is in italics, it means that the relevant option/field is currently not supported by SCOS-2000 but may be it will be in the future. Note that italics is also used to emphasize some important remarks reported in the descriptive text outside the tables definition.
- If the cell is in a normal font, the interpretation of the attribute is defined and it will be used by SCOS-2000 for processing.
- When a whole table is colored in grey, it means that the ASCII table is not required for the processing within SCOS-2000 and thus it will not be imported. However, it provides a suggestion for client mission about how the relevant table should be structured in order to support the implementation of mission specific applications (this is intended to harmonise the implementation of mission specific elements).
- All numeric values are to be expressed in decimal unless specified otherwise.
- An 'M' in the final column indicates a mandatory field i.e. a field which cannot be left Null.
- Fields which are not mandatory may or may not be explicitly given a value. Optional or unused fields which are left Null shall anyway be considered in that the field separator character corresponding to that field shall be present.
- The default value in the final column is the one assumed during the import in case an optional field is left Null.



Remark: This value should not be confused with the default value that the off-line database editors may adopt for each optional or mandatory field.

3.3.1 General

3.3.1.1 Database version: vdf

This table defines the database version identifier and is imported at the database generation time. If the table contains more than one record, the last imported one will determine the internal database version name and description. The database version name is used for display purposes and to ensure that the same database version has been used e.g. to generate a saved stack file and to load it.

Remark: note that the VDF table is mandatory i.e. the vdf.dat file must be present in the import directory with at least one record in it.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1	VDF_NAME	Char(8)	The name of the database version (any alphanumeric string of up to 8 chars).	M
2	VDF_COMMENT	Char(32)	Comment associated to the database version.	

3.3.2 Monitoring Data

In the following sections, the tables containing data related to the definition and processing of monitoring parameters, telemetry packets and monitoring displays are described.

A graphical overview of the database tables related to the definition of monitoring data and their relationships is shown in Appendix B. The following is a brief description of the monitoring data organisation.

The PCF is the central table in the definition of monitoring data. It specifies the characteristics of monitoring parameters and has the following main relationships with other tables:

- A many-to-1 relationship with the PCF table itself for the definition of state validity criteria. Each parameter may only reference one validity parameter, but the same validity parameter can be used by many parameters.
- A many-to-1 relationship with numeric calibration curves (CAF). Each parameter may only reference one curve, but each curve is shareable amongst other parameters. Each curve contains 1 or more points (CAP).
- A many-to-1 relationship with textual calibrations (TXF). Each parameter may only reference one textual calibration, but each set is shareable amongst other parameters. Each alias set contains 1 or more text points (PAS).
- A 1-to-1 relationship with monitoring checks definition (OCF). Each parameter may only reference one set of monitoring checks. However, each set may contain up to 16 different monitoring checks with their own type and applicability criteria.
- A 1-to-many relationship with the checks applicability criteria (OCP). Each monitoring check may reference one parameter to specify its applicability criteria. The same parameter can be re-used by many applicability criteria.
- A many-to-many relationship with telemetry packets (PID). Each parameter may be contained in zero or many packets. The same packet contains many parameters. The location of parameters within telemetry packets is specified in the PLF table for packets with a fixed content and in the VPD table for packets with a variable content.

- A many-to-many relationship with monitoring displays (DPF, GPF and SPF). Each parameter may be contained in zero or many displays. The same displays contains many parameters and possibly many times the same parameter. The content of monitoring displays is defined in the DPC, GPC and SPC tables.

3.3.2.1 Monitoring Parameters

The following sections describe the tables related to the definition of monitoring parameters i.e. telemetry parameters, user defined constants and synthetic parameters.

3.3.2.1.1 Monitoring parameters characteristics: pcf

This table defines the characteristics of monitoring parameters. One record per monitoring parameter.

Remark: note that SCOS-2000 automatically takes into account the validity of the validity expression associated to a monitoring parameter i.e. the state validity of a parameter is considered as TRUE only if the validity criteria is met AND the validity parameter is valid. Further, in case no instance of a periodic packet is received for a configurable interval, all parameters contained in it are automatically flagged as invalid ('expired'). This feature removes the need to include as part of the validity criteria all the conditions in the on-board status required to ensure that a packet is actually generated and transmitted.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PCF_NAME	Char(8)	Name of the parameter. Alphanumeric string uniquely identifying the monitoring parameter. Note that the OL syntax uses the start character(s) of OL components in order to identify their nature. As a consequence, no monitoring parameter name is allowed to start with the strings "VAR" or "GVAR" or "\$".	M
2	PCF_DESCR	Char(16)	Parameter Description. Free textual description of the parameter.	
3	PCF_PID	Number(10)	On-board ID of the telemetry parameter. This field allows the establishment of a one-to-one correspondence between on ground parameter ID and on-board "Parameter#" identifier (see [RD-6]). It is used to identify the parameter for which values are being delivered in a PUS compatible variable packet. The on-board parameter ID shall be unique i.e. it is not allowed to associate several telemetry parameters to the same on-board ID. In case two parameters are associated to the same PCF_PID, the last imported one will be used by SCOS-2000. Value is to be left null when there is no corresponding on-board parameter identifier, e.g 'ground only' parameters defined by the user.	

Fi. Nr	Field name	Field Type	Description	Ma/Def
4	PCF_UNIT	Char(4)	Engineering unit mnemonic of the parameter values e.g. 'VOLT'. This string is shown in monitoring displays besides the calibrated values for all parameters which are associated to a calibration (PCF_CURTX not null) or besides the raw values for all parameters which are not associated to any calibration (PCF_CURTX null) and are not unsigned integer (PCF_PTC > 3), typically real parameters.	
5	PCF_PTC	Number(2)	Parameter Type Code. This controls the encoding format of the parameter. It is used by SCOS-2000 in order to decode the value of parameters extracted from TM packets (PCF_NATUR = 'R' or PCF_NATUR = 'C'). A valid PTC value corresponding to the format in which the parameter value is calculated (e.g. integer, real, string) has to be specified for synthetic parameters as well (otherwise the parameter would be rejected at import time). Saved synthetic parameters (PCF_NATUR='S') must be given a PTC=13. Integer value in the range (1....13), see Appendix A.	M
6	PCF_PFC	Number(5)	Parameter Format Code. Along with the Parameter Type Code (PCF_PTC) this field controls the length of the parameter. It is only used by SCOS-2000 for parameters extracted from TM packets (PCF_NATUR = 'R' or PCF_NATUR = 'C'). However, a valid PFC value has to be specified for synthetic parameters as well (otherwise the parameter would be rejected at import time). Integer value in a range compatible with the specified PTC as specified in Appendix A.	M
7	PCF_WIDTH	Number(6)	'Padded' width of this parameter expressed in number of bits. This field is only used when extracting parameter samples using the VPD definition (see Section 3.3.2.5.2 below). If specified, it is used by the SCOS-2000 VPD application to identify the bit position where the next telemetry parameter starts. This can be useful in the case that the slot allocated to a parameter does not necessarily correspond to the parameter length implied by the parameter PTC/PFC pair e.g. in case of padding bits introduced in order to byte align the next parameter value. In particular it is useful for parameters of type deduced (PCF_PTC=11) in case they are allocated a fixed width (as opposed to a variable width depending on the type of the parameter value being downlinked). Note that this field is not used to determine the length of parameters extracted from fixed TM packets using the PLF definition (this is implicit in the definition of the parameters PTC and PFC, see Appendix A).	

Fi. Nr	Field name	Field Type	Description	Ma/Def
8	PCF_VALID	Char(8)	<p>Name of the parameter to be used to determine the state validity of the parameter specified in this record. Only parameters of type integer (signed or unsigned) can be used as validity parameters. The field PCF_VALPAR provides the validity value to be checked against. Note that SCOS-2000 automatically takes the validity of the validity expression (PCF_VALID = PCF_VALPAR) into account to evaluate the parameter validity.</p> <p>In case this is a Synthetic Parameter (PCF_NATUR not equal to 'R'), the overall state validity of the parameter will be the logical AND of the state validity of the calculated result (as derived from the state validity of the contributing parameters) and the state validity criteria explicitly associated to the SP (i.e. PCF_VALID = PCF_VALPAR).</p> <p>Warning: the 'view' of a monitoring parameter is updated every time that its validity condition is evaluated. This leads to an endless loop in case circular dependencies are established between validity parameters. For example, it must be avoided that a monitoring parameter is used as (or contributes to a Synthetic Parameter used as) the validity parameter of itself or even to the validity parameter of its validity parameter and so on.</p>	
9	PCF_RELATED	Char(8)	<p>Name of a monitoring parameter to be used for one of the two following purposes:</p> <p>For Saved Synthetic Parameters (PCF_NATUR = 'S'), this field has to be used to identify the name of 'original' synthetic parameter, whose computed value is used for the archival within synthetic parameter packets.</p> <p>For telemetry parameters of type 'deduced' (PCF_PTC = 11), this field has to be used to identify the name of the telemetry parameter (referred to as Parameter# in [RD-6]) whose value provides the on-board parameter ID. The on-board parameter ID is used to interpret the parameters of type 'deduced' in variable TM packets (see Section 3.3.2.5.2 below).</p>	
10	PCF_CATEG	Char(1)	<p>Calibration category of the parameter.</p> <p>'N' - for Numerical. This option shall be used for parameters associated to a numerical or polynomial calibration curve. Also numerical parameters which are not associated to any calibration curve should use this option.</p> <p>'S' - for Status. This option should only be used for parameters associated to a textual calibration curve.</p> <p>'T' - for Text. This option should only be used for parameters of type Character String (PCF_PTC = 8).</p>	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
11	PCF_NATUR	Char(1)	<p>Nature of the parameter.</p> <p>'R' – Raw telemetry parameter (i.e. a monitoring parameter whose value is extracted from TM packets),</p> <p>'D' – Dynamic OL parameter (i.e. a synthetic parameter specified in Operations Language and not compiled. One ASCII file containing the OL expression associated to this parameter must exist, see Section 3.3.2.1.2 below),</p> <p>'H' – Hard Coded parameter (i.e. a synthetic parameter which has been directly specified in C++ or a synthetic parameter which was initially defined in OL and eventually compiled, see Section 3.3.2.1.2 below),</p> <p>'S' – Saved Synthetic parameter (i.e. a synthetic parameter whose value is calculated based on the expression associated to the parameter specified in PCF_RELATED. The calculated value is then saved in synthetic packets, see Section 3.3.2.1.2 below),</p> <p>'C' – Constant parameter (also referred to as 'static' User Defined Constant i.e. a user defined parameter for which a static value is specified in the field PCF_PARVAL). It is not allowed to associate this parameter also to a PLF entry describing its position in the 'constants TM packet' (see Section 3.3.2.1.3 below).</p>	M
12	PCF_CURTX	Number(4)	<p>Parameter calibration identification number. Depending on parameter category, this field stores the numerical calibration curve identification number or the textual calibration identification number.</p> <p>Integer value matching with TXF_NUMBR (if PCF_CATEG = 'S') or with CAF_NUMBR/MCF_IDENT of the corresponding calibration (textual, numerical or polynomial, respectively).</p> <p>This field cannot be null for status parameters (PCF_CATEG = 'S'). It must be left null for textual parameters (PCF_CATEG = 'T'), for string parameters (PCF_PTC = 7 or 8) and for time parameters (PCF_PTC = 9 or 10).</p>	

Fi. Nr	Field name	Field Type	Description	Ma/Def
13	PCF_INTER	Char(1)	<p>Flag controlling the extrapolation behaviour for parameters calibrated using a numerical calibration curve.</p> <p>'P' – using this option, if a raw value outside the calibration curve is received, a valid engineering value is calculated by extrapolating the first two calibration points (in case the raw value is outside the calibration range on the lower side) or the last two calibration points (in case the raw value is outside the calibration range on the upper side)</p> <p>'F' – using this option, if a raw value outside the calibration curve is received, an invalid engineering value is returned.</p> <p>This field is only relevant if PCF_CATEG='N' and PCF_CURTX is not null.</p> <p>In case of textual calibration (i.e. PCF_CATEG = 'S'), if a raw value not associated to any text string is received, an invalid engineering value set to all stars '*****' is returned.</p> <p>Note that SCOS-2000 associates the extrapolation flag to the calibration curve specified in PCF_CURTX and not to each individual parameter. This implies that, in case several parameters use the same calibration curve, they will all be displayed using the value of PCF_INTER for the last imported one. In case two different extrapolation flags are required to be used, two distinct calibration curves have to be defined.</p>	
14	PCF_USCON	Char(1)	<p>Flag controlling the execution of status consistency checks for this parameter.</p> <p>'Y' - if the parameter has to be status consistency checked</p> <p>'N' - otherwise.</p> <p>This field must be set to 'Y', if the OCP table (see Section 3.3.2.3.2 below) contains a status consistency entry for this parameter.</p>	'N'
15	PCF_DECIM	Number(3)	Number of decimal places to be used for displaying real values of this monitoring parameter.	
16	PCF_PARVAL	Char(14)	<p>Default raw value of the constant parameter. Only applicable if the parameter is a 'static' UDC (PCF_NATUR = 'C'), in which case it is a mandatory field.</p> <p>The value must be expressed in a format compatible with the parameter type (as determined by the PCF_PTC and PCF_PFC, see Appendix A).</p>	
17	PCF_SUBSYS	Char(8)	<p>Name of the group to which the monitoring parameter belongs. This field can be used in the future to organise parameters for filtering purposes (future extension).</p>	
18	PCF_VALPAR	Number(5)	Raw value of the validity parameter (PCF_VALID). This is used to evaluate the state validity expression (PCF_VALID = PCF_VALPAR).	'1'

Fi. Nr	Field name	Field Type	Description	Ma/Def
19	PCF_SPTYPE	Char(1)	Output view of the synthetic parameter calculation. 'E' – Engineering i.e. it cannot be associated to a calibration curve. 'R' – Raw.	

3.3.2.1.2 Synthetic parameters

Synthetic parameters are monitoring parameters whose source value is calculated by SCOS-2000 (these were referred to as derived parameters in previous control systems). As specified in Section 3.3.2.1.1 above, SCOS-2000 supports the following three categories of synthetic parameters:

- Dynamic OL Synthetic Parameters (PCF_NATUR = 'D'): these parameters are associated to an Operations Language definition. The syntax to be used is documented in [RD-9]. The dynamic OL synthetic parameters are evaluated locally and only if required by any server process or user application (e.g. by the TM desktop in order to display the value of the synthetic parameter itself in an AND or to evaluate the validity of another displayed parameter). If required, the dynamic OL synthetic parameters are evaluated every time that a sample of any of the contributing 'dynamic' parameters (see [RD-9]) is received and its value or validity has changed. The calculated values are not stored. When in retrieval mode, dynamic OL synthetic parameters are also re-calculated using the same approach as for real-time. Each dynamic OL synthetic parameter has to be associated to a dedicated ASCII file containing its definition. The file can contain any number of lines. The file name has to be the name of the synthetic parameter itself (without any file extension). The SCOS-2000 importer will read in the text contained within each synthetic parameter file and verify its content. After import, it is possible to modify on-line the OL definition. This on-line change only affects the local processing unless it is 'broadcasted' (in which case the new definition is used by all processes/applications requiring it until the next database import).

Remark: in case a super-commutated parameter is used in the definition of a synthetic parameter, this will be calculated once for each sample of the contributing parameter. This may turn out to be extremely demanding in terms of resources usage. Including super-commutated parameters in the definition of synthetic parameter shall be therefore carefully considered.

- Hard-coded Synthetic Parameters (PCF_NATUR = 'H'): these parameters are associated to a C++ file which can be generated either directly or by using the OL compiler (to convert a dynamic OL Synthetic Parameter into a hard-coded one, see [RD-9]). The C++ files associated to hard-coded synthetic parameters have to be linked to the relevant library. They are used by SCOS-2000 exactly in the same way as described above for dynamic OL synthetic parameters. The advantages of using hard-coded synthetic parameters (as opposed to dynamic OL synthetic parameters) are that they are much less demanding in terms of processing power and also that they can be directly specified in C++ thus overcoming any restriction in the supported OL syntax. The disadvantage of hard-coded synthetic parameters is that their definition cannot be visualised and modified on-line.
- Saved Synthetic Parameters (PCF_NATUR = 'S'): these parameters are linked to another synthetic parameter (either dynamic OL or hard-coded) via the PCF_RELATED field. Their value is calculated by a centralised process named SPPG (Synthetic Parameters Packet Generator) based on the definition of the associated synthetic parameter. Values are calculated systematically every time that a 'changed' sample of a contributing parameter is received. However, the Synthetic Packets are generated only when a specified TM packet is received. The list of Synthetic Packets (identified by their SCOS-2000 Packet ID) has to be configured in the file sppg.apid.ctrl (typically under \home\sops1\resources). The SCOS-2000 Packet ID of the packet triggering the generation of a Synthetic Packet is specified in the file sppg.map (also typically under \home\sops1\resources). The calculated values are stored as calculated (i.e. in raw or engineering form) along with the validity flags in dedicated SCOS-2000 TM packets (Synthetic Packets) which are generated by the SPPG process.

Remark: in case a super-commutated parameter is used in the definition of the synthetic parameter associated to a saved synthetic parameter, this will be calculated once for each sample of the contributing parameter. However, only one value will be stored in the Synthetic Packet. This basically implies that it is not possible to define super-commutated saved synthetic parameters.

Each saved synthetic parameter requires to be associated to a PLF entry specifying its position in the related Synthetic Packet. This position is used by the SPPG to generate the Synthetic Packet and also by any other application to read the value of the saved synthetic parameter. This implies that changes in the position of a saved synthetic parameter in a synthetic packet shall be avoided across different database versions (otherwise the application retrieving the stored values will read them in a position which is not the same one used by the SPPG at the time of the synthetic packet generation). Each saved synthetic parameter shall be associated to an entry in the PLF for the associated Synthetic Packet (no overlap is allowed). It has to be given a length of 13 bytes (8 bytes are used to store the parameter value, one byte is used to store the value type and 4 bytes are used to store the validity flags). It is possible to define several Synthetic Packets in the PLF table, in order to group saved synthetic parameters in a meaningful manner (e.g. all synthetic parameters related to one TM packet are stored together in a single Synthetic Packet). The advantages of saved synthetic parameters are that they are only calculated centrally (i.e. they do not imply any overhead in the local TM processing applications) and are not re-calculated in retrieval mode. This minimises the impact on the workstation performance and also allows the display of the parameter value as it was in real-time (independently of any subsequent change in the definition of the associated synthetic parameter). Finally, saved synthetic parameters are the only ones ensuring that the same value is displayed for a given time when navigating in retrieval forwards or backwards. The disadvantage of saved synthetic parameters is that not necessarily all the calculated values are stored. In fact, a Synthetic packet is only generated once the associated TM Packet is received and not every time that a changed value for any contained parameter is received/calculated.

Apart from the obvious difference in the way that the source value is calculated, synthetic parameters are handled by SCOS-2000 in the same way as telemetry parameters. Synthetic parameters can be displayed in any monitoring display, they can be limit checked, they can be used in the definition of other synthetic parameters (by referring to the .syn view, see [RD-9]), etc. Synthetic parameter values can also be calibrated. In this case SCOS-2000 applies the associated calibration curve taking the synthetic parameter source value as the raw value (independently on whether the source value has an engineering meaning or is actually a raw value). The validity of the synthetic parameter source value is implicitly calculated by SCOS-2000 based on the current validity of the parameter values actually used for its computation. It is possible to further restrict the validity of synthetic parameters by explicitly associating a validity condition in the PCF_VALID/PCF_VALPAR fields.

Remark: it should be noted that SCOS-2000 synthetic parameters are potentially calculated every time that any contributing sample is received. This is fundamentally different than previous control systems where the calculation of derived parameters was triggered by the reception of a specified source TM packet. The implication of the SCOS-2000 approach is that circular dependencies between synthetic parameters must be avoided. In fact, take the case of parameter C which contains parameter B which contains parameter A which contains parameter C. If any sample contributing to e.g. parameter A is received, this triggers the calculation of parameter A, B and C in cascade. The calculation of parameter C on its turn triggers the calculation of parameter A again, thus entering into an indefinite loop!

More detailed guidelines about how to define and best use the different categories of synthetic parameters are provided in the Synthetic Parameters SUM, see [RD-9].

3.3.2.1.3 User Defined Constants

This section briefly describes the handling of User Defined Constants (UDCs) within SCOS-2000. It is intended to be used as a guide for the definition of the relevant database entries.

The following two categories of User Defined Constants are supported by SCOS-2000:

1. The 'static UDCs' i.e. constants which are associated to a default value specified in the database which can only be changed (either locally and temporarily or globally and permanently) using the OL editor of the Telemetry Query Display. In fact, static UDCs are handled by SCOS-2000 as synthetic parameters with a constant value. The constant value used for static UDCs is initialised based on the default value specified in the database unless any permanent change has been applied to it since the last database import (see [RD-15]).
2. The 'dynamic UDCs' i.e. constants whose value can be specified on-line via TM SPACON. These are handled by SCOS-2000 as standard raw TM parameters whose value is extracted/stored into an internally generated TM packet.

Remark: previous SCOS-2000 releases allowed to associate UDCs to both the static and the dynamic view. As from R2.3, this needs to be avoided as the OL syntax only supports one view to return the parameter source value independently of its nature (see [RD-9]).

For each User Defined Constant (independently on its category) a PCF record shall be created specifying its characteristics exactly in the same way as for a standard telemetry parameter. The field PCF_NATUR shall be set to 'C' if the parameter has to be associated to the static default value ('static UDC', category 1 above), the constant value being specified in the field PCF_PARVAL. The field PCF_NATUR shall be set to 'R' if the parameter is associated to the dynamic view ('dynamic UDC', category 2 above).

The TM SPACON application provides the ability to modify on-line the values associated to each 'dynamic UDC'. This is based on the generation of a SCOS-2000 TM packet containing the latest values for all dynamic UDCs (the SCOS-2000 Packet ID of this 'constants TM packet' is configurable in the MISCconfig SCOS-2000 initialisation file). Dynamic User Defined Constants shall thus be associated to a PLF entry enabling the generation and the extraction of the updated UDC value out of the 'constants TM packet'. No overlap between the bit ranges covered by different dynamic UDCs is allowed. However, 'holes' in the definition of the 'constants TM packet' are allowed (i.e. two consecutive PLF entries of this packet do not necessarily need to be contiguous).

The UDC dynamic value is used to initialise the relevant TM SPACON display and is distributed (as part of the 'constants TM packet') to all server and client applications. The updated UDC values (both for static and dynamic UDCs) are processed exactly in the same way as any other telemetry parameter value i.e. it can be calibrated, it can be OOL checked, it can be used as validity or applicability parameter, etc.

When using UDCs as part of Operations Language expression, the following shall be considered:

- in order to use the user specified (source) value of both static and dynamic UDCs, the '. raw' parameter view has to be used (e.g. C1000.raw).
- in order to use the calibrated value of both static and dynamic UDCs, the '. eng' parameter view has to be used (e.g. C1000.eng). This is only allowed for UDCs associated to a calibration curve (PCF_CURTX not null).

The following table summarizes the characteristics of UDCs belonging to the different categories listed above.

Category	PCF_NATUR	PLF Record	Static Value Source	Dynamic Value Source
1 (static UDC)	C	No	PCF_PARVAL	OL Editor Broadcast Packet
2 (dynamic UDC)	R	Yes	None	Constants TM Packet

Remark: it should be noted that only UDCs of a numerical nature (integer or real) or character strings are supported i.e. no time, no octet strings.

3.3.2.1.4 Dynamic Configuration Variables

This section briefly describes the handling of Dynamic Configuration Variables within SCOS-2000. It is intended to be used as a guide for the definition of the relevant database entries.

The MISC Server process maintains the status of the Configuration Variables which can be dynamically changed and distributes any update to all its clients. In order to enable the monitoring of the variables status using the standard monitoring processing, the MISC Server automatically generates a telemetry packet (referred to as the 'MISC' packet) containing the current value of selected Configuration Variables. The SCOS-2000 Packet ID of this 'MISC' TM packet is configurable in the MISCconfig SCOS-2000 initialisation file.

The content of the 'MISC' packet can be used in the same way as any other TM packet e.g. for display purposes, remote monitoring via external interfaces, applicability criteria, limit checking, pre-transmission validation, command execution verification.

Warning: only Configuration Variables that are encoded as integer or real values can be fully processed by the standard monitoring applications. The Configuration Variables that are encoded as character strings can only be displayed.

This section provides guidelines for the definition of the database entries (in the PCF and PLF tables) which are required to extract the data encoded into the 'MISC' TM packet generated by the MISC Server process. The following applies:

- Dynamic Configuration Variables (see the Configuration and Installation Guide, [RD-1]) can be associated in the MISCconfig file to a monitoring parameter which is used to encode and distribute its value as 'pseudo-telemetry'
- For each Configuration Variable associated to a monitoring parameter, a PCF record shall be created specifying its characteristics exactly in the same way as for a standard telemetry parameter
- The PCF entry corresponding to the Configuration Variable shall be given an appropriate type (PCF_PTC) and length (PTC_PFC) as specified in the Configuration and Installation Guide, see [RD-1]. Note that all Configuration Variables are to be specified as 32-bit Signed Integers (PCF_PTC=4 and PCF_PFC=14), Double Precision Reals (PCF_PTC=5 and PCF_PFC=2) or Character Strings (PCF_PTC=8 with length depending on the specific variable).
- In addition, each monitoring parameter associated to a Configuration Variable must also appear in a PLF entry with the SPID of the 'MISC' TM packet. The PLF entries are used by the MISC Server in order to encode (and extract at 'warm' initialization time) the current value of the Configuration Variable in the 'MISC' TM packet
- No overlap between the bit ranges covered by different monitoring parameters is allowed.
- 'Holes' in the definition of the 'MISC' packet are allowed (i.e. two consecutive PLF entries of this packet do not necessarily need to be contiguous).
- The position of the same monitoring parameter shall be kept consistent across subsequent database versions (this is required in order to properly process older 'MISC' packets in the history files).

3.3.2.2 Monitoring parameters calibration

The following sections define the tables required to perform calibration of monitoring parameter values. Calibrations can be applied to any type of monitoring parameters (i.e. raw telemetry parameters, user defined constants as well as synthetic parameters). For synthetic parameters, the source value i.e. the value returned by the associated OL or hard-coded definition is used as the raw value. Only parameters which in their raw representation are of a numerical nature can be associated to a calibration (i.e. no strings, times).

3.3.2.2.1 Numerical calibration curves: caf

This table defines the numerical calibration curves corresponding to one or more monitoring parameters. One record per calibration curve.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CAF_NUMBR	Number(4)	Numerical calibration curve identification number. Integer value in the range (1...9999) uniquely identifying the numerical calibration curve.	M
2	CAF_DESCR	Char(32)	Textual description of the calibration curve	
3	CAF_ENGFMT	Char(1)	Flag identifying the format type for the engineering values specified in the corresponding records of the CAP table (see Section 3.3.2.2.2 below). ‘I’ – for signed Integer ‘U’- for Unsigned integer ‘R’- for Real	M
4	CAF_RAWFMT	Char(1)	Flag identifying the format type for the raw values (CAP_XVALS) specified in the corresponding records of the CAP table (see Section 3.3.2.2.2 below). ‘I’ – signed integer ‘U’- unsigned integer ‘R’- real This format has to be compatible with the type of the parameters associated to this calibration curve.	M
5	CAF_RADIX	Char(1)	Flag identifying the radix used for the raw values specified in the corresponding records of the CAP table (see Section 3.3.2.2.2 below). Only applicable for unsigned integer values (i.e. CAF_RAWFMT = ‘U’). ‘D’ – Decimal ‘H’ – Hexadecimal ‘O’ – Octal	
6	CAF_UNIT	Char(4)	Engineering unit mnemonic. Available for consistency checking against the unit of any associated monitoring parameter (PCF_UNIT).	

Fi. Nr	Field name	Field Type	Description	Ma/Def
7	CAF_NCURVE	Number(3)	Number of points defined in the CAP table for this curve.	

3.3.2.2.2 Numerical calibration curves definition: cap

This table contains the numerical value pairs defining the monitoring calibration curves. One record per raw/engineering value pair.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CAP_NUMBR	Number(4)	Calibration curve identification number. Integer value matching with CAF_NUMBR of the corresponding calibration curve.	M
2 *	CAP_XVALS	Char(14)	Raw value of the calibration point. It has to be expressed in a format compatible with CAF_RAWFMT. Unsigned integer values (CAF_RAWFMT = 'U') have to be expressed using the radix associated to this calibration curve (CAF_RADIX).	M
3	CAP_YVALS	Char(14)	Engineering value of the calibration point. It has to be expressed in a format compatible with CAF_ENGFMT. All values including unsigned integer values have to be expressed in decimal regardless of the radix associated to this calibration curve (CAF_RADIX).	M

3.3.2.2.3 Textual calibrations: txf

This table defines the textual calibrations (also known as aliases sets) corresponding to one or more monitoring parameters. One record per textual calibration.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	TXF_NUMBR	Number(4)	Textual calibration identification number. Integer value in the range (1...9999) uniquely identifying the textual calibration.	M
2	TXF_DESCR	Char(32)	Description of the textual calibration.	

Fi. Nr	Field name	Field Type	Description	Ma/Def
3	TXF_RAWFMT	Char(1)	Flag identifying the format type for the raw values specified in the corresponding records of the TXP table (see Section 3.3.2.2.4 below). 'I' – signed integer 'U'- unsigned integer 'R'- real This format has to be compatible with the type of the parameters associated to this textual calibration.	M
4	TXF_NALIAS	Number(3)	Number of records defined in the TXP table for this textual calibration.	

3.3.2.2.4 Textual calibrations definition: txp

This table defines the text strings corresponding to one or more parameter raw values. One record per values/string pair.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	TXP_NUMBR	Number(4)	Textual calibration identification number. Integer value matching with TXF_NUMBR of the corresponding textual calibration.	M
2 *	TXP_FROM	Char(14)	Lower range raw value. It has to be expressed (in decimal) in a format compatible with TXF_RAWFMT.	M
3	TXP_TO	Char(14)	Upper range raw value. It has to be expressed (in decimal) in a format compatible with TXF_RAWFMT.	M
4	TXP_ALTXT	Char(14)	Alphanumeric string displayed as engineering value corresponding to any parameter raw value falling in the range (TXP_FROM....TXP_TO)	M

3.3.2.2.5 Polynomial calibration curves: mcf

This table contains the coefficients associated to the polynomial calibration curves. One record per polynomial calibration curve. SCOS-2000 will calculate the engineering value 'Y' corresponding to the raw value 'X' of a parameter using the following formula:

$$Y = A_0 + A_1 * X + A_2 * X^2 + A_3 * X^3 + A_4 * X^4$$

Remark: note that the identification number of polynomial calibration curves cannot overlap with the identification numbers of numerical calibration curves (CAF_NUMBR). This can be achieved by e.g. reserving a range of curved identification numbers for polynomial calibrations.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	MCF_IDENT	Number(4)	Polynomial calibration curve identification number. Integer number in the range (1...9999) uniquely identifying the polynomial calibration.	M
2	MCF_DESCR	Char(32)	Description of the polynomial calibration	
3	MCF_POL1	Char(14)	Polynomial coefficient A ₀ . Numerical value expressed in real format (see Appendix A).	M
4	MCF_POL2	Char(14)	Polynomial coefficient of the 1 st order (A ₁) Numerical value expressed in real format (see Appendix A).	'0'
5	MCF_POL3	Char(14)	Polynomial coefficient of the 2 nd order (A ₂) Numerical value expressed in real format (see Appendix A).	'0'
6	MCF_POL4	Char(14)	Polynomial coefficient of the 3 rd order (A ₃) Numerical value expressed in real format (see Appendix A).	'0'
7	MCF_POL5	Char(14)	Polynomial coefficient of the 4 th order (A ₄). Numerical value expressed in real format (see Appendix A).	'0'

3.3.2.3 Monitoring parameter checks

3.3.2.3.1 Monitoring checks: ocf

This table defines for each parameter name, the characteristics of the monitoring checks to be applied to it. At most one record per parameter.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	OCF_NAME	Char(8)	Name of the parameter. Alphanumeric string matching with PCF_NAME of the parameter to be checked.	M
2	OCF_NBCHCK	Number(2)	Number of consecutive valid parameter samples violating the check required to declare the parameter as OOL.	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
3	OCF_NBOOL	Number(2)	Number of checks associated to this parameter in the OCP table (see Section 3.3.2.3.2 below). Integer number in the range (1....16) It must be consistent with the number of records specified in the OCP table for this parameter.	M
4	OCF_INTER	Char(1)	Flag identifying the interpretation of the limit values (OCP_LVALU and OCP_HVALU) specified in the OCP table for this parameter. 'U' - Uncalibrated, if raw values are used. 'C' - Calibrated, if engineering values are used. Option 'C' can only be used for parameters associated to a calibration curve (PCF_CURTX not null).	M
5	OCF_CODIN	Char(1)	Flag identifying the interpretation of the limit values (OCP_LVALU and OCP_HVALU) specified in the OCP table for this parameter. 'R' - Real values. This option can only be used for raw values of real parameters (OCF_INTER = 'U' and PCF_PTC = 5) or for engineering values of numerically calibrated parameter (OCF_INTER = 'C' and CAF_ENGFMT = 'R'). 'A' - ASCII string, This option 'A' can only be used for status parameters (PCF_CATEG = 'S') if the expected values are expressed in engineering form (OCF_INTER='U'). 'I' - Integer values (otherwise).	M

3.3.2.3.2 Monitoring checks definition: ocp

This table contains the definition of the monitoring checks associated to each monitoring parameter specified in the OCF table. The checks associated to a parameter are executed by SCOS-2000 in the order in which they are specified in this table. Each check is associated to an applicability criteria which enables its execution (OCP_RLCHK = OCP_VALPAR). On the top of the traditional monitoring checks, SCOS-2000 support also checks of type Event (OCP_TYPE = 'E') are not meant to be used for monitoring purposes. These checks are to be used as a mechanism to trigger the generation of SCOS-2000 event packets based on the detection of a specific condition in the telemetry data. As soon as the monitored parameter goes out of the specified range (provided that the check is enabled by its applicability criteria), SCOS-2000 automatically generates an event with the ID "BEHV::OCP_NAME" (where OCP_NAME is the name of the parameter to which the check is associated). The generation of this event can be used to trigger system actions as described in [RD-10] (e.g. send an e-mail, set the value of a SCOS-2000 configuration variable, execute a script file). The monitoring parameter triggering the generation of this event will not be shown in the OOL display.

The following rules for the population/processing of OCP records apply:

- Checks associated to a parameter are checked for applicability in the order of import (the table is supposed to be sorted in an appropriate order based on OCP_NAME and OCP_POS).

- The checks applicability criteria (parameter OCP_RLCHK valid and equal to OCP_VALPAR) are used to determine the applicability of all checks except Status Consistency checks (OCP_TYPE='C') for which the applicability criteria is forced to TRUE.
- The identification of applicable checks for a parameter stops as soon as an applicable check or a Soft/Hard pair of applicable checks is encountered. This applies also to the checks of type Status Consistency (OCP_TYPE='C') or Events (OCP_TYPE='E'). It is possible to configure the system such that all applicable checks are applied in any case but this can only be done on a global basis (see MISConfig variable BEHV_STOP_FIRST_SET) i.e. not on an individual basis per parameter.
- Checks of type Status Consistency (OCP_TYPE='C') or Events (OCP_TYPE='E') must always precede other checks for the same parameter (if any).
- Soft/Hard pairs of monitoring checks must be associated to the same applicability criteria and must follow each other in the order Soft/Hard. It is allowed to have the same low or high limit value for Soft and Hard limits associated to the same applicability criteria.
- A Soft OOL check (OCF_TYPE='S') can be specified in isolation i.e. it does not necessarily require the existence of a Hard OOL check associated to the same applicability criteria.
- It is possible to create sets of allowed values for status checks (i.e. the ones applied to parameters associated to PCF_CATEG='S'). This is achieved by inserting several OCP records with the same type (Soft or Hard) containing the different allowed values in OCP_LVALU. The applicability criteria associated to the first imported record with that type is the used one.
- A Soft or Hard check is violated if the telemetry is outside the allowed range of values. In case the telemetry value matches exactly one of the limit values, this is not considered as an OOL condition.
- Parameters violating several checks (e.g. a Soft and a Hard check) are displayed according to the violated check with the higher criticality.

Remark: The OCP table is expected to be sorted in OCP_NAME and OCP_POS order.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	OCP_NAME	Char(8)	Name of the parameter. Alphanumeric string matching with OCF_NAME of the parameter to be checked.	M
2 *	OCP_POS	Number(2)	Used to define the order in which the checks are to be applied for each parameter. The OCP table is expected to be sorted by OCP_NAME and OCP_POS, with the pairs ordered by soft and hard entries.	

Fi. Nr	Field name	Field Type	Description	Ma/Def
3	OCP_TYPE	Char(1)	Flag identifying the type of monitoring check. 'S' – Soft OOL 'H' – Hard OOL 'C' – Status Consistency (in which case the field PCF_USCON for the parameter OCP_NAME must be set to 'Y') 'E' - Event generation only (no OOL).	M
4	OCP_LVALU	Char(14)	Parameter value to be expressed in a format compatible with OCF_CODIN. It is used as the low limit value (for numerical parameters i.e. if PCF_CATEG = 'N') or as the expected status value (for status parameters i.e. if PCF_CATEG = 'S'). This field is irrelevant for Status Consistency checks (OCP_TYPE = 'C').	
5	OCP_HVALU	Char(14)	High limit value to be expressed in a format compatible with OCF_CODIN (see Appendix A). This field is only relevant for checks associated to numerical parameters (PCF_CATEG = 'N'). It is irrelevant for checks associated to status parameters (PCF_CATEG='S') and for Status Consistency checks (OCP_TYPE = 'C').	
6	OCP_RLCHK	Char(8)	Name of the parameter to be used to determine the applicability of this monitoring check. Only parameters of type integer (signed or unsigned) can be used as applicability parameters. The field OCP_VALPAR provides the applicability value to be checked against. Note that SCOS-2000 automatically takes the validity of the applicability expression (OCP_RLCHK = OCP_VALPAR) into account to evaluate the check applicability. This field is not used for Status Consistency checks (OCP_TYPE='C').	
7	OCP_VALPAR	Number(5)	Raw value of the applicability parameter (OCP_RLCHK). This is used to evaluate the check applicability expression (OCP_RLCHK = OCP_VALPAR). This field is not used for Status Consistency checks (OCP_TYPE='C').	'1'

3.3.2.4 Telemetry Packets Definition

The following sections describe the tables related to the processing of incoming telemetry packets and the generation of SCOS-2000 packets by the telemetry packetiser. The PID and PIC tables specify the general characteristics of incoming TM packets and the criteria to recognize and uniquely identify them within SCOS-2000. The characteristics of the SCOS-2000 TM Packets are specified in the tpcf table.

3.3.2.4.1 Packets identification: pid

This table contains the information specifying the correspondence between the SCOS-2000 Telemetry Packet Number (a.k.a. the SCOS-2000 Packet ID) and the source packets generated by the spacecraft. This table is used by the telemetry packetiser in order to identify the incoming packets and enable the further processing of them.

The structure of the SCOS-2000 TM packets (and thus the way to extract parameters out of them, see Section 3.3.2.5 below) is uniquely identified by the SCOS-2000 Packet ID (SPID) for fixed packets and by the Telemetry Packet Structure Definition (TPSD) for variable packets.

Remark: it is possible to process the same TM packet using both the standard TM processing (PLF definition) and the Variable Packets Display (VPD definition).

The key to identify a packet is composed by the first 5 fields of the table, i.e. TYPE, STYPE, APID, PI1_VAL and PI2_VAL. It is allowed to associate different keys to the same SPID. It is also possible to have more records with the same key, and in that case the field VALID determines the SPID to be considered. For each key there shall be (at most) only one valid instance.

Remark: The allocation between packets identification fields (APID/Type/Subtype/PI1/PI2) and S2K SPIDs should not change after starting the archiving of mission telemetry in the history files. Further, PID entries containing SPIDs which are present in the history files shall not be deleted. This is required in order to ensure proper functioning of some advanced data retrieval services (such as the Generic Data Distribution System).

Remark: Although this is not in principle imposed by the structure of the PID table, there shall only exist one valid combination of SPID/TPSD/APID/Type/Subtype (basically any two TM packets which only differ in the additional identification fields PI1 and PI2 shall not be associated to the same combination of processing keys SPID and TPSD). This is required in order to support proper filtering of the SCOS-2000 Packets History Display.

Remark: it is recommended that also the internally generated TM packets (i.e. packets which are generated by SCOS-2000 applications as opposed to the ones received as part of the spacecraft telemetry) are associated to a PID entry. This is desirable as some SCOS-2000 applications (e.g. the Packets History Display) use the PID definition in order to derive the list of SPIDs to be retrieved in order to match a specified packet filter. The PID entry corresponding to internally generated TM packets (e.g. the User Defined Constant TM packet, see section 3.3.2.1.3 above, the MISC TM packet see section 3.3.2.1.4 above, the Synthetic Packet see section 3.3.2.1.2 above) must be associated to a Type/Subtype/APID combination which is not expected to be received from any other source. A TPCF entry related to the internal TM packets is also required in order to display the proper packet mnemonic in the monitoring displays.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PID_TYPE	Number(3)	Type of the TM source packet (see [RD-6]) Integer number in the range (0...255).	M
2 *	PID_STYPE	Number(3)	Subtype of the TM source packet (see [RD-6]) Integer number in the range (0...255).	M
3 *	PID_APID	Number(5)	APID of the TM source packet. Integer number in the range (0...2047) for TM source packets generated by the spacecraft (see [RD-6]) and in the range (2048...65535) for any other SCOS-2000 TM packet (e.g. TM packets generated by SCOS-2000 applications or by other control domains such as SCOE equipments).	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
4 *	PID_PI1_VAL	Number(10)	Value of the first packet additional identification field (e.g. value of Structure ID, Task ID, etc.). It is used, together with the packet APID, Type and Sub-type, to identify the Telemetry Packet structure and to interpret its content (if relevant, see Section 3.3.2.4.2 below). Unsigned integer number.	'0'
5 *	PID_PI2_VAL	Number(10)	Value of the second packet additional identification field (e.g. value of Function ID, etc.). It is used, together with the packet APID, Type, Sub-type and first additional field, to identify the Telemetry Packet structure and to interpret its content (if relevant, see Section 3.3.2.4.2 below). Unsigned integer number.	'0'
6 *	PID_SPID	Number(10)	The SCOS-2000 Telemetry Packet Number (also referred to as the SCOS-2000 Packet Id). This field uniquely identifies the structure of TM packets defined in the PLF (see Section 3.3.2.5.1 below) and determines the history file in which this packet is archived. Unsigned integer number greater than zero (zero is not allowed).	M
7	PID_DESCR	Char(64)	Textual description of Telemetry Packet. Where applicable (see field PID_EVENT below), it is used by the telemetry packetiser in order to associate an appropriate message text to the SCOS-2000 Event Packets generated on reception of this telemetry source packet.	
8	PID_UNIT	Char(8)	<i>On-board subsystem. This field could be used to group packets in a meaningful manner for filtering purposes (future extension).</i>	
9	PID_TPSD	Number(10)	The SCOS-2000 Telemetry Packet Structure Definition. This field is only used by the Variable Packets Display application. It has to be set to '-1' for packets which are not defined in the VPD table and thus are not required to be processed by the Variable Packets Display. It has to be set to the packet structure identification number for a packet which is defined in the VPD table (the TPSD is used by the Variable Packets Display application in order to determine how to process this packet, see Section 3.3.2.5.2 below). Integer number.	-1
10	PID_DFHSIZE	Number(2)	This field is used in order to provide a start reference for the processing of variable packets (see Section 3.3.2.5.2 below). Typically, it corresponds to the size of the source TM packet header plus the packet data field header. This field is irrelevant in case of fixed packets (PID_TPSD= '-1'). Integer number of octets in the range (0...99).	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
11	PID_TIME	Char(1)	Flag identifying the presence of the packet generation time field in the TM source packet. 'Y' - if time field is present 'N' - if time field is not present.	'N'
12	PID_INTER	Number(8)	The packet generation interval for periodic TM packets expressed in number of milliseconds. It is used in order to identify the time interval after which the parameter samples received in a periodic packet are declared as expired. This field must be left null for non-periodic packets.	
13	PID_VALID	Char(1)	Flag indicating whether the combination of key field shall be considered as valid. This field can be used to identify the SPID to be used in case more records with the same key exist. 'Y' - if the combination is valid 'N' - if the combination is invalid. At import time, if more records with the same key have this field set to 'Y,' only the last one will be considered as valid. The other ones will be force to have PID_VALID='N'.	'Y'
14	PID_CHECK	Number(1)	This field is used in order to indicate the presence of the Packet Error Control field (checksum) in the TM source packet and whether and how this should be checked by the telemetry packetiser when an instance of this packet is received. '0' - if the telemetry packetiser shall not check the Packet Error Control field (either because this TM Source Packet does not contain or because this check shall be disabled for this packet) '1' - if the telemetry packetiser shall check the Packet Error Control field by applying the Cyclic Redundancy Code (CRC) as specified in [RD-6]. Additional options to support other 'check-summing' algorithms can be added by the client missions if needed.	'0'

Fi. Nr	Field name	Field Type	Description	Ma/Def
15	PID_EVENT	Char(1)	<p>This field is used to indicate whether the telemetry packetiser shall generate a SCOS -2000 Event Packet when receiving this TM source packet and also to determine its severity. This mechanism can be used to notify the operators of the arrival of specified packets (e.g. anomaly events). Note that the message text of the SCOS-2000 Event Packet is extracted out of the field PID_DESCR (see above).</p> <p>'N' - if no SCOS-2000 Event Packet has to be generated 'I' - if a SCOS -2000 Event Packet of severity Info shall be generated 'W' - if a SCOS-2000 Event Packet of severity Warning shall be generated 'A' - if a SCOS -2000 Event Packet of severity Alarm shall be generated.</p>	'N'
15	PID_EVID	Char(17)	<p>An alphanumeric string used by the telemetry packetiser as Event ID for the generation of the SCOS-2000 Event Packet. Associating an Event ID to a TM Source Packet can be useful in order to control the recipients of the SCOS-2000 Event Packet and to trigger a specified action on reception of this packet. This field contains the Event ID proper (used for controlling the distribution of the Event) and optionally also the Action ID (used in order to trigger a specified action) in the following format (see [RD-10]): [Application Event id]:[Application Action ID]</p> <p>This field is optional and only relevant for packets triggering the generation of a SCOS-2000 Event Packet (i.e. if PID_EVENT is not set to 'N').</p>	

3.3.2.4.2 Packets identification criteria: pic

This table specifies which additional identification fields have to be used in order to uniquely identify an incoming TM packet. For each packet type/subtype combination, the relevance and the position of the additional identification fields is specified. This table is directly loaded by the SCOS-2000 telemetry packetiser i.e. it is not imported into the run-time database.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PIC_TYPE	Number(3)	Type of the TM source packet (see [RD-6]) Integer number in the range (0....255).	M
2 *	PIC_STYPE	Number(3)	Sub-type of the TM source packet (see [RD-6]) Integer number in the range (0....255).	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
3	PIC_PI1_OFF	Number(5)	This field identifies the offset of the first identification field starting from the beginning of the TM source packet (i.e. the beginning of the SCOS-2000 packet body) . Integer number of bytes (value to be set to -1 if no additional identification field is necessary for this packet type/subtype combination).	M
4	PIC_PI1_WID	Number(3)	Width of the first additional packet identification field expressed in number of bits.	M
5	PIC_PI2_OFF	Number(5)	This field identifies the offset of the second identification field (if any) starting from the beginning of the TM source packet (i.e. the end of the SCOS-2000 packet body) . Integer number of bytes (value to be set to -1 if the second additional identification field is not relevant to this packet type/subtype combination).	
6	PIC_PI2_WID	Number(3)	Width of the second additional packet identification field expressed in number of bits.	

3.3.2.4.3 Telemetry packets characteristics: tpcf

This table contains the characteristics of SCOS-2000 TM Packets i.e. all attributes which are associated to all TM Packets identified by the same SPID (e.g. the packet mnemonic which is used in order to provide the user with a more meaningful identification of the processed packets). At most one record per SCOS-2000 Packet ID.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	TPCF_SPID	Number(10)	The SCOS-2000 Telemetry Packet Number (also referred to as the SCOS-2000 Packet Id). Unsigned integer number matching with PID_SPID of the packet.	M
2	TPCF_NAME	Char(12)	The alphanumeric mnemonic of the Telemetry Packet. It is used in the monitoring displays to identify the source packet of parameter values. Note: In case no mnemonic is specified in the database for a packet to be processed, the string 'None' will be displayed in the relevant field of the monitoring displays.	
3	TPCF_SIZE	Number(8)	The total length of the telemetry packet expressed in bytes. This field has to be set to zero for variable length packets. This field is available to mission specific applications e.g. to generate the History Files configuration file based on this table.	

3.3.2.5 Telemetry Packets Processing

The following sections describe the tables related to the extraction of telemetry parameter values from telemetry packets. The PLF and VPD tables specify the structure of telemetry packets with a fixed or a variable content, respectively.

Note that also telemetry packets internally generated by SCOS-2000 or mission specific applications will be processed in the same way as standard telemetry packets. This implies that PLF or VPD entries shall be defined in the database for them. This applies e.g. to the 'constant packet' (as described in Section 3.3.2.1.3 above) or to the 'MISC Packet' (as specified in Section 3.3.2.1.4 above).

Remark: only the parameters contained within packets which are defined in the VPD table are displayed in full mode of the SCOS-2000 Packets History Displays.

3.3.2.5.1 Parameters location in fixed packets: plf

This table details the layout of telemetry packets which have a fixed format. Telemetry parameters can appear in several packets at the same time. 'Overlapping' entries (i.e. two parameters extracted from the same packet portion) as well as 'gaps' (i.e. packets portion from where no parameter value is extracted) are allowed. The only restriction is that it is not allowed to have more than one record per TM parameter in the same packet (supercommutated parameters are defined using a single record).

The standard monitoring displays such as ANDs, GRDs and SCDs will make use of the information contained in the PLF table to extract parameter values out of the received packets.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PLF_NAME	Char(8)	Name of the parameter. Alphanumeric string matching with PCF_NAME of the monitoring parameter to be extracted.	M
2 *	PLF_SPID	Number(10)	Telemetry Packet Number. Positive integer value matching with SCOS-2000 Packet ID (PID_SPID) of the TM packet out of which the parameter values are extracted.	M
3	PLF_OFFBY	Number(5)	Location of first occurrence of parameter value in octets, relative to the end of SCOS-2000 TM header (not to be confused with the TM source packet header). Integer value starting from 0 (negative values are not allowed).	M
4	PLF_OFFBI	Number(1)	Bit number, within an octet, of the first bit of the first occurrence of the parameter value. Integer value in the range (0...7).	M
5	PLF_NBOCC	Number(4)	Number of occurrences of the parameter in the packet. Integer value in the range (1...9999). 1 has to be used for non-supercommutated parameter	'1'
6	PLF_LGOCC	Number(5)	Number of bits between the start of 2 consecutive occurrences (irrelevant if not supercommutated i.e. if PLF_NBOCC = 1). Integer value in the range (0...32767)	'0'

Fi. Nr	Field name	Field Type	Description	Ma/Def
7	PLF_TIME	Number(9)	Time offset of first parameter occurrence relative to packet time (in milliseconds). This field is used to adjust the time stamp of the extracted value for each individual parameter occurrence. This equally applies to parameters which are super-commutated or not. Integer value in the range (-4080000...4080000).	'0'
8	PLF_TDOCC	Number(9)	Time delay (in milliseconds) between 2 consecutive occurrences (irrelevant if not supercommutated i.e. if PLF_NBOCC = 1). Integer value in the range (0...4080000).	'0'

Note 1: PLF_TIME and PLF_TDOCC are used by the display subsystem to plot super commutated parameter values in graphic displays. They are also used to display times on scrolling displays and printouts.

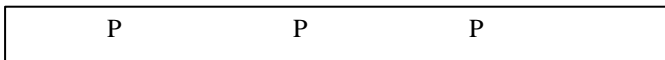
Note 2: To fully understand the different types of parameters, please consider the following definitions and examples.

Let be P a parameter, the following 2 cases are possible:

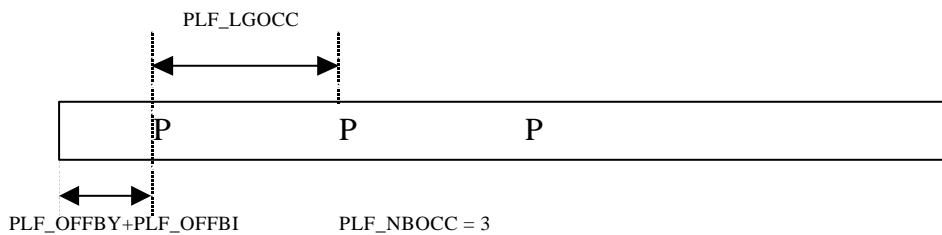
Case 1) P is non commutated (PLF_NBOCC = 1). It appears once in the packet



Case 2) P is supercommutated (PLF_NBOCC > 1). It appears periodically several times in the packet



The figure below illustrates the fields used for extracting all samples of supercommutated parameters:



3.3.2.5.2 Variable packet definition: vpd

This table contains the definition of variable TM packets. This table allows the description of any TM packet containing data according to the PUS structure rules (see [RD-6]). The VPD has no relation to the DPF, GPF, SPF or PLF table. The VPD will be used for packet processing and display together with the PCF and its related tables.

The VPD table can also be used to define particular fixed-structure packets, in order to use the Variable Packet Display application for their visualisation. It should be noted that this kind of packets, containing no repeated groups, would be displayed by the Variable Packet Display on a single line, as a sequence of parameters. The VPD_NEWLINE field allows the visualisation of such packets using a line for each parameter, displaying its name, description and value as specified in VPD_DCHAR.

Remark: the definition of variable packets in the VPD table allows gaps and overlaps of parameters. The location of each parameter in the packet is in fact calculated based on the end of the previous parameter corrected by the applicable offset (positive offsets allow the introduction of gaps in the packet definition whereby negative offsets allow the introduction of overlaps i.e. the use of the same bits for the extraction of consecutive parameters). In addition, padding bits can be handled by an appropriate definition of the parameter width in the PCF (see description of field PCF_WIDTH in Section 3.3.2.1.1 above). Padding bits are defined as unused bits which always precede the parameter value (i.e. most significant bits) in the TM packet.

Remark: The VPD is expected to be sorted in VPD_TPSD and VPD_POS order.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	VPD_TPSD	Number(10)	Telemetry Packet Structure Definition. Integer number matching with PID_TPSD of the telemetry packet to be interpreted.	M
2 *	VPD_POS	Number(4)	Ordinal position of this parameter inside the packet definition. To be used by the database editors to deliver the records in the ASCII table in the appropriate order.	
3	VPD_NAME	Char(8)	Name of the parameter appearing in the variable telemetry packet. Alphanumeric string matching with PCF_NAME of the monitoring parameter to be processed/displayed.	M
4	VPD_GRPSIZE	Number(3)	This value should only be set for parameters which identify a repeat counter (typically referred to as 'N' in [RD-6]) or for 'dummy' parameters (see field VPD_FIXREP below) which identify a fixed number of repetitions (see VPD_FIXREP below). It specifies the number of parameters in the group to be repeated. In other words, the parameters group repeated by this counter is defined by the successive 'GRPSIZE' records of the VPD table.	'0'
5	VPD_FIXREP	Number(3)	This value should only be set for parameters which identify a repeat counter for arrays of fixed size. When set, this value defines the fixed number of repetitions for the group of parameters determined by VPD_GRPSIZE. Note that parameters for which this value is set are to be defined in the PCF though they are not present in this variable packet. They serve the purpose of identifying the fixed number of repetitions of the array (whose size is identified in VPD_GRPSIZE).	'0'

Fi. Nr	Field name	Field Type	Description	Ma/Def
6	VPD_CHOICE	Char(1)	Flag identifying whether this parameter is to be used to determine the structure of the following portion of the packet. 'Y' - if value obtained from TM packet is used as key (TPSD) to interpret the following entries in the packet. 'N' - if this field is not of choice type.	'N'
7	VPD_PIDREF	Char(1)	Flag indicating whether the value delivered in this field provides the on-board identifier of a telemetry parameter (typically referred to as 'Parameter#' in [RD-6]). The value is compared against existing PCF_PID values in order to identify the corresponding SCOS-2000 name of the telemetry parameter. 'Y' - if value is a parameter id. 'N' - if value is not a parameter id.	'N'
8	VPD_DISDESC	Char(16)	Textual description of the parameter value. It will be used on the displays for the generation of the header line.	
9	VPD_WIDTH	Number(2)	Field width used to control whether the parameter value is displayed. 0 - means that the value is not displayed. >0 - any number greater than zero implies that the value is displayed with its full length.	M
10	VPD_JUSTIFY	Char(1)	Flag controlling the justification used in the display for the value field. 'L' - Left justified 'R' - Right justified 'C' - Centered	'L'
11	VPD_NEWLINE	Char(1)	Flag controlling whether the parameter shall be displayed on a new line. 'Y' - display/print value on new line. 'N' - display/print value within same line. NOTE: The value 'Y' is only allowed for parameters that are not part of structures containing repeated groups. For repeated groups there is an automatic alignment to display each repetition on a single row. When set to 'Y', all the parameters of the same packet are displayed on a different line.	'N'

Fi. Nr	Field name	Field Type	Description	Ma/Def
12	VPD_DCHAR	Number(1)	Flag controlling which data are displayed for the parameter. 0 – Display only the value 1 – Display VPD_NAME and value 2 – Display VPD_NAME, PCF_DESCR and value This field is only relevant for packets containing a fixed number of parameters (i.e. no group repeater). For variable packets the value '2' can only be used for parameters with VPD_PIDREF='Y', in order to display the parameter description PCF_DESCR (as well the parameter name PCF_NAME). For all other parameters only the value is displayed, since the description is in the header of its column.	'0'
13	VPD_FORM	Char(1)	Display format of parameter. 'B' – Binary (only applicable for raw values of unsigned integer parameters) 'O' – Octal (only applicable for raw values of unsigned integer parameters) 'D' – Decimal 'H' – Hexadecimal (only applicable for raw values of unsigned integer parameters) 'N' – Normal (i.e. the engineering value is displayed for parameters associated to a calibration curve or the raw value in decimal for the other ones).	'N'
14	VPD_OFFSET	Number(6)	Number of bits between the start position of this parameter and the end bit of the previous parameter in the packet. A positive offset enables the introduction of a 'gap' between the previous parameter and this one. A negative offset enables the 'overlap' of the bits contributing to this parameter with the ones contributing to the previous parameter(s). Integer value in the range (-32768...32767)	'0'

Below are some examples detailing the usage of the VPD table. The VPD_OFFSET column is omitted as in the given examples the applicable offset is systematically zero (no gaps and no overlaps).

The next two tables detail the definition of the PUS Out-of-limit List report (12,11) (see [RD-6]) and its possible representation within the VPD table.

N	Parameter#	Parameter Value	Limit Crossed	Previous Check	Current Check	Transition Time
Unsigned	Enumerated	Deduced	Deduced	Enumerated	Enumerated	Time
Integer						

TPSD	POS	NAME	GRP-SIZE	FIX-REP	CHOICE	PID-REF	DISDEC	WIDTH	JUS-TIFY	NEW-LINE	DCHAR	FORM
10	1	Number	6	0	N	N	Nr. of OOL	0	L	N	0	N
10	2	ParRef	0	0	N	Y	ParamName	6	L	N	0	N
10	3	ParVal	0	0	N	N	Value	14	L	N	0	N
10	4	LimitCrS	0	0	N	N	LimitCrossed	14	L	N	0	N
10	5	PrvCheck	0	0	N	N	PreviousChecking	14	L	N	0	N
10	6	CurCheck	0	0	N	N	CurrentChecking	14	L	N	0	N
10	7	TranTime	0	0	N	N	TransitionTime	21	L	N	0	N

The next four tables illustrate the definition of the structured choice type and its possible representation within the VPD table. The first table represents the choice type being received within a telemetry packet. The following two tables are possible representations of choice types being referenced by the Choice field of the first table. If the telemetry packet contains as the choice value '11', then the structure definition of (N,A,B,C) is used, while in case of '12' (N,A,B) is used.

Field1	Field2	Choice
Unsigned Integer	Bool	Enumerated

N	A	B	C
Unsigned Integer	Unsigned Integer	Integer	Bool

N	A	B
Unsigned Integer	Unsigned Integer	Integer

TPSD	POS	NAME	GRP-SIZE	FIX-REP	CHOICE	PID-REF	DISDEC	WIDTH	JUS-TIFY	NEW-LINE	DCHAR	FORM
1	1	Field1	0	0	N	N	Field1	14	L	N	0	N
1	2	Field2	0	0	N	N	Field2	14	L	N	0	N
1	3	Choice	0	0	Y	N	Choice	0	L	N	0	N
11	1	Number	3	0	N	N	Number	0	L	N	0	N
11	2	A	0	0	N	N	A	14	L	N	0	N
11	3	B	0	0	N	N	B	14	L	N	0	N

TPSD	POS	NAME	GRP-SIZE	FIX-REP	CHOICE	PID-REF	DISDEC	WIDTH	JUS-TIFY	NEW-LINE	DCHAR	FORM
11	4	C	0	0	N	N	C	14	L	N	0	N
12	1	Number	2	0	N	N	Number	0	L	N	0	N
12	2	A	0	0	N	N	A	14	L	N	0	N
12	3	B	0	0	N	N	B	14	L	N	0	N

Note that in the group of parameters with the same TPSD, the choice parameter must be the last one, since the next parameter to be extracted from the packet will be determined depending on the structure identified by the value of the choice parameter.

The next example shows how a fixed array structure (composed of the parameters B and C repeated 6 times) is represented in a VPD table.

TPSD	POS	NAME	GRP-SIZE	FIX-REP	CHOICE	PID-REF	DISDEC	WIDTH	JUS-TIFY	NEW-LINE	DCHAR	FORM
1	1	A	0	0	N	N	Field1	14	L	N	0	N
1	2	N	2	6	N	N	Counter	0	L	N	0	N
1	3	B	0	0	N	N	Field2	14	L	N	0	N
1	4	C	0	0	N	N	Field3	14	L	N	0	N

Note that in order to represent the fixed array structure, an additional field (N) is needed in the VPD table. It specifies the size of the group to be repeated (in this case GRPSIZE=2 to repeat the following two parameters, B and C) and the fixed number of times the group is to be repeated (in this case FIXREP=6).

3.3.2.6 Displays and reports

The following sections describe the tables related to the definition of monitoring displays (Alphanumeric, Graphical and Scrolling) and reports (printouts). Note that the values of parameters appearing on these displays are only the ones extracted from fixed packets using the PLF table. The values extracted out of TM variable packets can only be displayed/printed using the VPD application.

3.3.2.6.1 Alphanumeric displays: dpf

This table contains the list of Alphanumeric Displays (32 or 64 parameter displays). One record per display proforma.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	DPF_NUMBE	Char(8)	Name of the Alphanumeric Display (AND). Alphanumeric string uniquely identifying the AND.	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
2	DPF_TYPE	Char(1)	Type of alphanumeric display '1' - 32-parameter display with information about the packet contained the last displayed parameter sample '3' - 64-parameters display.	M
3	DPF_HEAD	Char(32)	The alphanumeric header that will appear on the top line of the display.	

3.3.2.6.2 Alphanumeric displays definition: dpc

This table contains the list of parameters to be displayed in each AND and their position. One record per displayed parameter.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	DPC_NUMBE	Char(8)	Name of the AND. Alphanumeric string matching with DPF_NUMBE of the corresponding AND.	M
2	DPC_NAME	Char(8)	Name of the monitoring parameter to be displayed (PCF_NAME).	M
3 *	DPC_FLDN	Number(2)	Parameter position in the display (0,63). Positions (0,31) are displayed on the left side, positions (32-63) are displayed on the right side.	M
4	DPC_COMM	Number(4)	Commutation of the parameters to be displayed. Numeric value in the range (1....9999), left justified.	'1'
5	DPC_MODE	Char(1)	Flag enabling/disabling the effect of the Telemetry Desktop configuration option controlling the display of invalid parameter values. 'Y' – invalid values only displayed if the relevant configuration option in the Telemetry Desktop is set to 'Display invalid values' 'N' – invalid values always displayed (the display of invalid values is forced independently of the setting of the relevant configuration option in the Telemetry Desktop).	'Y'

Fi. Nr	Field name	Field Type	Description	Ma/Def
6	DPC_FORM	Char(1)	Display format of parameter. 'B' – Binary (only applicable for raw values of unsigned integer parameters) 'O' - Octal (only applicable for raw values of unsigned integer parameters) 'D' – Decimal 'H' – Hexadecimal (only applicable for raw values of unsigned integer parameters) 'N' – Normal (i.e. the engineering value is displayed for parameters associated to a calibration curve or the raw value in decimal for the other ones).	'N'

3.3.2.6.3 Graphic displays: gpf

This table contains the list of Graphic Displays (GRDs). One record per graphical display.

Remark: this file as well as the GPC table (see below) do not get imported into SCOS-2000 database but are directly read by SCOS-2000 applications.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	GPF_NUMBE	Char(8)	Name of the Graphical Display (GRD). Alphanumeric string uniquely identifying the GRD.	M
2	GPF_TYPE	Char(1)	Type of graphic display 'F' - Full screen, 8 plots against time or another parameter. 'H' - Half screen, 2 times 4 plots. 'Q' - Quarter screen, 4 times 2 plots. 'S' - Sliced screen, 8 times 1 plot. Note: the specified number of plots per screen portion is a maximum number, presently 8.	M
3	GPF_HEAD	Char(32)	The alphanumeric header that will appear on the top line of the display.	
4	GPF_SCROL	Char(1)	Flag controlling the scrolling of the display when reaching the end of the plot area(s). Meaningless for parameters plotted against spin phase or another parameter. 'Y' - for scrolling 'N' - otherwise.	'N'

Fi. Nr	Field name	Field Type	Description	Ma/Def
5	GPF_HCOPY	Char(1)	Flag controlling the auto hardcopy at the end of plot area. 'Y'- for auto-hardcopy 'N'- otherwise	'N'
6	GPF_DAYS	Number(2)	Number of days for the time span of X-axis. For parameter against parameter display, it is used to determine the duration of the display. Integer value in the range (0...99)	M
7	GPF_HOURS	Number(2)	Number of hours for the time span of X-axis. For parameter against parameter display, it is used to determine the duration of the display. Integer value in the range (0...23)	M
8	GPF_MINUT	Number(2)	Number of minutes for the time span of X-axis. For parameter against parameter display, it is used to determine the duration of the display. This field can be expressed as a fraction of minutes, e.g. 0.1 means that the X-axis will be of a duration of 6 seconds (if GPF_DAYS=GPF_HOURS=0). Numerical number in the range (0.0...59.9)	M
9	GPF_AXCLR	Char(1)	Color of axes, ticks and grids. '1' - Green '2' - Blue '3' - Cyan '4' - Red '5' - Yellow '6' - Magenta '7' - White	M
10	GPF_XTICK	Number(2)	Number of intervals between ticks on X-axis Integer value in the range (1...99)	M
11	GPF_YTICK	Number(2)	Number of intervals between ticks on Y-axis Integer value in the range (1...99)	M
12	GPF_XGRID	Number(2)	Number of intervals between grids on X-axis Integer value in the range (1...99)	M
13	GPF_YGRID	Number(2)	Number of intervals between grids on Y-axis Integer value in the range (1...99)	M
14	GPF_UPUN	Number(2)	Display update rate in milliseconds. Available for external applications accessing the database. Not used by SCOS-2000.	

3.3.2.6.4 Graphic displays definition: gpc

This table contains the list of parameters to be displayed in each GRD. One record per displayed parameter.

Remark: The GPC is expected to be sorted in GCP_NUMBE and GPC_POS order.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	GPC_NUMBE	Char(8)	Name of the GRD. Alphanumeric string matching with GPF_NUMBE of the corresponding GRD.	M
2 *	GPC_POS	Number(1)	Order of parameter entries in the GRD.	
3	GPC_WHERE	Char(1)	Identification of the plot area where the parameter is to be plotted. The number of plot areas depends on the GRD type (GPF_TYPE) as detailed below. For GRDs of type 'F' (Full screen): '1' - if parameter to be plotted 'P' - if X-axis parameter For GRDs of type 'H' (Half screen): '1' - top graph '2' - bottom graph For GRDs of type 'Q' (Quarter screen): areas are clockwise numbered (1....4) starting with the top-left plot area For GRDS of type 'S' (Sliced screen): areas are top-down numbered (1....8) starting with the top-most plot area.	M
4	GPC_NAME	Char(8)	Name of the monitoring parameter to be displayed in the plot (PCF_NAME).	M
5	GPC_RAW	Char(1)	Flag controlling whether the calibrated or raw value shall be plotted. 'C' or space – the calibrated value will be plotted. 'U' – otherwise	'U'
6	GPC_MINIM	Char(14)	Minimum value for the parameter. This limit will be associated to the start of the axis on which the parameter value is projected. The value shall be specified in raw or engineering form depending on GCP_RAW in a compatible format (see Appendix A).	M
7	GPC_MAXIM	Char(14)	Maximum value for the parameter. This limit will be associated to the end of the axis on which the parameter value is projected. The value shall be specified in raw or engineering form depending on GCP_RAW in a compatible format (see Appendix A).	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
8	GPC_PRCLR	Char(1)	Color of the symbol and connecting lines. '1' - Green '2' - Blue '3' - Cyan '4' - Red '5' - Yellow '6' - Magenta '7' - White	M
9	GPC_SYMB0	Char(1)	Symbol to be plotted for each parameter value. '0' - corresponding to no symbol '1' - corresponding to X '2' - corresponding to + '3' - corresponding to 0 '4' - corresponding to * '5' - corresponding to <> '6' - corresponding to []	'0'
10	GPC_LINE	Char(1)	Line type of the lines connecting 2 parameter values (i.e. 2 symbols). '0' - corresponding to no line '1' - corresponding to solid line '2' - corresponding to long dash '3' - corresponding to dash '4' - corresponding to short dash '5' - corresponding to dash alternated with short dash.	'0'

3.3.2.6.5 Scrolling displays: spf

This table contains the list of Scrolling Displays (SCDs). One record per scrolling display.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	SPF_NUMBE	Char(8)	Name of the Scrolling Display (SCD). Alphanumeric string uniquely identifying the SCD.	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
2	SPF_HEAD	Char(32)	The alphanumeric header that will appear on the top line of the display.	
3	SPF_NPAR	Number(1)	The number of parameters in the display. It must match with the number of records associated to this SCD in the SPC table (see Section 3.3.2.6.6 below). Integer value in the range (1....5).	M
4	SPF_UPUN	Number(2)	Display update rate in milliseconds. Available for external applications accessing the database. Not used by SCOS-2000.	

3.3.2.6.6 Scrolling displays definition: spc

This table contains the list of parameters to be displayed in each SCD. One record per displayed parameter.

Remark: The SPC is expected to be sorted in SPC_NUMBE and SPC_POS order.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	SPC_NUMBE	Char(8)	Name of the SCD. Alphanumeric string matching with SPF_NUMBE of the corresponding SCD.	M
2 *	SPC_POS	Number(1)	Order of entries in the scrolling display.	
3	SPC_NAME	Char(8)	Name of monitoring parameter contributing to the scrolling display (PCF_NAME).	M
4	SPC_UPDT	Char(1)	Flag controlling whether the parameter is to be updated by scrolling onto a new line, or by overwriting the old value with the new one. ' ' (Space or Null) - updating by line scrolling 'N' - updating by overwriting (no scrolling).	' '
5	SPC_MODE	Char(1)	Flag enabling/disabling the effect of the Telemetry Desktop configuration option controlling the display of invalid parameter values. ' ' (Space or Null) - invalid values only displayed if the relevant configuration option in the Telemetry Desktop is set to 'Display invalid values' 'N' - invalid values always displayed (the display of invalid values is forced independently of the setting of the relevant configuration option in the Telemetry Desktop).	' '

Fi. Nr	Field name	Field Type	Description	Ma/Def
6	SPC_FORM	Char(1)	Format used to display the parameter value 'B' - Binary (only applicable for raw values of unsigned integer parameters) 'O' - Octal (only applicable for raw values of unsigned integer parameters) 'D' - Decimal 'H' - Hexadecimal (only applicable for raw values of unsigned integer parameters) 'N' - Normal (i.e. the engineering value is displayed for parameters associated to a calibration curve or the raw value in decimal for the other ones).	'N'
7	SPC_BACK	Char(1)	Background colour used for displaying parameter field. '0' - Black '1' - Green '2' - Blue '3' - Cyan '4' - Red '5' - Yellow '6' - Magenta '7' - White	'0'
8	SPC_FORE	Char(1)	Foreground colour used for displaying parameter field. '0' - Black '1' - Green '2' - Blue '3' - Cyan '4' - Red '5' - Yellow '6' - Magenta '7' - White	M

3.3.2.6.7 Printout proforma: ppf

This table will contain the list and the format specification of telemetry printout proforma. One record per proforma. The files defining the printout proforma (ppf and ppc) do not necessarily need to be imported into the run-time database. They can also be used as initialisation files which are directly accessed by the relevant application.

Remark: currently it is only possible to define printout proforma using the relevant SCOS-2000 editor. Extensions and modifications to the underlying data structures will be required in order to support the import of printout definitions as specified in the following tables.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PPF_NUMBE	Char(4)	Name of the printouts proforma. Alphanumeric character string uniquely identifying the printout proforma.	M
2	PPF_HEAD	Char(32)	A print header that will appear on the top line of the printed pages.	
3	PPF_NBPRO	Number(2)	Number of parameters to be printed. Integer value in the range (1....32). It must be consistent with the number of records specified in the PPC table for this printout.	

3.3.2.6.8 Printout proforma definition: ppc

This table contains the list of parameters to be printed in each printout proforma. One record per monitoring parameter belonging to the printout.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PPC_NUMBE	Char(4)	Name of the printout proforma. Alphanumeric string matching with PPF_NUMBE of the corresponding printout.	M
2 *	PPC_POS	Number(2)	Parameter position in the printout.	M
3	PPC_NAME	Char(8)	Name of the monitoring parameter to be printed as part of the printout (PCF_NAME).	M
4	PPC_FORM	Char(1)	Format used to print the parameter values. 'B' – Binary (only applicable for raw values of unsigned integer parameters) 'O' - Octal (only applicable for raw values of unsigned integer parameters) 'D' - Decimal 'H' - Hexadecimal (only applicable for raw values of unsigned integer parameters) 'N' - Normal (i.e. the engineering value is displayed for parameters associated to a calibration curve or the raw value in decimal for the other ones).	'N'

3.3.3 Commanding data

Appendix B shows an overview of the commanding tables structure and the relationships between the various database entities. The following is a brief description of the commanding data organisation. Many-to-many relationships by their nature require there to be an intermediate file which maps the two together (e.g. the PSM or the CVP).

The most important entity in the database is the command definition (CCF) which describes the contents of a command. A command has the following main relationships:

- A 1-to-many relationship with PTV expressions. Each PTV expression (PTV) can only be referenced by one command, but a command can reference many expressions, each of which are ANDed together to form the validation criteria. If an expression is to be repeated for another command, it must be defined again.
- A many-to-many relationship with verification stages (CVS). This is because a command can reference many stages, but a stage is also shareable between many commands (therefore stages need only be defined once - as they contain a lot of information). A mapping table (CVP) is required which supports the relationship. Each stage based on monitoring parameters contains one or more expressions (CVE) which are logically ANDed together to form the verification criteria. The CVE table can also be used to specify the monitoring parameters which are affected by each command execution stage (used for status consistency checking).
- A many-to-many relationship with parameter sets (PST). Again parameter value sets can be shared and a command may reference more than one. Therefore a mapping table (PSM) is required. Each parameter set contains one or more parameter value sets (PVS).
- A many-to-many relationship with command parameters (CPC). The mapping table (CDF) defines the location of each parameter instance in the application data field of any command.

A command parameter (CPC) describes an individual part of a command's contents, which define its characteristics and behaviour once released to the spacecraft. A command parameter has the following relationships:

- A many-to-1 relationship with numeric calibration curves (CCA). Each parameter may only reference one curve, but each curve is shareable amongst other parameters. Each curve contains 1 or more points (CCS).
- A many-to-1 relationship with textual calibrations (or alias set) (PAF). Each parameter may only reference one alias set, but each set is shareable amongst other parameters. Each alias set contains 1 or more text points (PAS).
- A many-to-1 relationship with range sets (PRF). Each parameter may only reference one range set, but each set is shareable amongst other parameters. Each range set contains 1 or more entries consisting of a min-max pair or a single discrete value (PRV).

A command sequence (CSF) contains an ordered list of elements (which can be commands or sequences themselves) which also contain dynamic release information allowing the sequence to be expanded (during load time on a command source) into individual command requests ready for dispatch. A command sequence has the following main relationships:

- A 1-to-many relationship with sequence elements (CSS) which can be commands or sequences. Hence, there is an implied relationship between the CSS and the CCF/CSF. Each element is also parameterised (SDF). There is a many-to-1 relationship between the SDF and the CDF/CSP tables. There is also a 1-to-1 relationship between an entry in the SDF and a parameter value set (PVS), see below.
- A many-to-many relationship with parameter sets (in the same way as for commands).
- A 1-to-many relationship with the parameters (CSP) which parameterise the sequence (the sequence parameters have been traditionally referred to as 'formal parameters'). They can be used to parameterise element parameters (which can be formal parameters themselves if the element is a sequence), or a sequence element itself (i.e. a non-parametric command or sequence). In this way a sequence is treated as a template containing place-holders

where sequence parameter values can be provided in order to control the behaviour of the sequence. This means fewer sequences need to be defined where their operation is similar.

A sequence parameter (CSP) is defined in a similar way to command parameters and has the same relationships with the calibration and range check definitions.

A TC packet (TCP) defines each packet header which, a command will be contained in once it is encoded prior to be released to the spacecraft. I.e. it will be pre-pended before the command parameters (application data). A command packet header has the following relationships:

- A many-to-many relationship with packet header parameters (PCPC). These define characteristics of a packet header element, for example header fields. Each packet may have many parameters, and parameters can be shared amongst other packets. This relationship is mapped by the (PCDF) file which, describes the location of each instance of a parameter to its associated packet header.
- A 1-to-many relationship with command definitions (CCF). Each command must reference a single packet, but the same packet may be referenced by other commands.

3.3.3.1 Packet Headers

The following tables define the characteristics and the structure of command packet headers. This is used by SCOS - 2000 in order to build the first 'portion' of the encoded TC packet. The packet header is followed by the command application data field which is defined via the CPC and CDF tables.

Remark: the description of the field CCF_CTYPE in Section 3.3.3.2.1 below specifies how to use the PCDF and CDF tables for on-board control directives.

3.3.3.1.1 Packet headers characteristics: tcp

This table defines the TC packet headers. One record per TC packet header.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	TCP_ID	Char(8)	Unique name of the packet header.	M
2	TCP_DESC	Char(24)	Textual description of the packet header.	

3.3.3.1.2 Packet headers parameters: pcpc

This table defines the TC packet header parameters. One record per packet header parameter.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PCPC_PNAME	Char(8)	Unique name of the packet header parameter.	M
2	PCPC_DESC	Char(24)	Textual description of the header parameter.	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
3	PCPC_CODE	Char(1)	Flag identifying the format in which the value is specified for this parameter in PCDF_VALUE. 'I' – signed integer 'U' – unsigned integer	'U'

3.3.3.1.3 Packet headers definition: pcdff

This table defines the structure of each packet header by referencing packet header parameters. One record per element constituting the packet header.

Remark: note that packet header parameters cannot be directly edited by the operators when loading commands on a SCOS-2000 manual stack. Their value is appropriately set by the command source or by the command handler (e.g. the APID, Type and Subtype are extracted from the relevant CCF fields for each command, the packet length is automatically set based on the actual packet length including repeated groups (see Section 3.3.3.2.3 below), the Source Sequence Counter is maintained by the multiplexer for each APID).

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PCDF_TCNAME	Char(8)	Name of the packet header (TCP_ID) to which this element belongs to.	M
2	PCDF_DESC	Char(24)	Textual description used for fixed areas only.	
3	PCDF_TYPE	Char(1)	Type of the header element: 'F' - fixed area whose value is taken from PCDF_VALUE. 'A' - APID Parameter whose value is to be taken from CCF_APID 'T' - Service Type Parameter whose value is to be taken from CCF_TYPE 'S' - Service Sub-Type Parameter whose value is to be taken from CCF_STYPE 'K' – Acknowledgement Flags Parameter whose value is to be taken from CCF_ACK 'P' - Other packet parameter whose value is automatically set by the SCOS-2000 command subsystem (e.g. packet length, source sequence counters).	M
4	PCDF_LEN	Number(4)	Length of the header element expressed in bits. Unsigned integer expressed in decimal. Note that all PCDF records with the same element name (PCDF_PNAME) must also be associated to the same element length (PCDF_LEN). This constraint applies whether the element is used several times within the same packet header or even within different packet headers (PCDF_TCNAME)	M

5 *	PCDF_BIT	Number(4)	Bit offset of the header element calculated from the start of the entire packet. Unsigned integer number of bits.	M
6	PCDF_PNAME	Char(8)	Name of the header parameter (PCPC_PNAME). Only applicable if type is 'A', 'T', 'S' or 'P'. It has to be left Null if type is 'F'.	
7	PCDF_VALUE	Char(10)	Value of the fixed area (PCDF_TYPE = 'F') expressed in Hex or the default value of the header element otherwise (expressed in decimal for signed integer parameters and in PCDF_RADIX for unsigned integer parameters).	M
8	PCDF_RADIX	Char(1)	Radix in which a parameter value (PCDF_VALUE) is given if unsigned integer (PCPC_CODE = 'U'). 'D' – Decimal 'H' – Hexadecimal 'O' – Octal. This field is irrelevant for parameters of type signed integer.	'H'

3.3.3.2 Commands

The following sections describe the tables related to the definition of commands (CCF table) and command parameters (CPC table). [The specific routine and destination of commands can be optionally specified using the DST table.](#) The structure of the command application data field is specified via the CDF table in a way which is compatible with the need to generate variable length commands (see [RD-6]). Each command element whose value requires editing and/or calibration and/or checking has to be defined as a command parameter. Command parameters can be re-used in many different command definitions.

Remark: only command parameter values (as opposed to fixed areas, see Section 3.3.3.2.3 below) are visible in the command history (fixed areas are only stored as part of the TC packet Hex dump).

Remark: the command used to load execution time-tagged commands (referred to as 'Insert Telecommands in the Command Schedule, (11,4)' in [RD-6]) is required to be created in the database but only its CCF definition (its CDF definition is not used as its encoding is based on a hard-coded structure).

3.3.3.2.1 Command characteristics: ccf

This table defines the characteristics of a command. One record per command.

Remark: commands can be destined to the spacecraft or to other control domains such as the ground test equipments.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CCF_CNAME	Char(8)	Unique name of the command.	M
2	CCF_DESCR	Char(24)	Textual description of the command.	M
3	CCF_DESCR2	Char(64)	Detailed description.	

Fi. Nr	Field name	Field Type	Description	Ma/Def
4	CCF_CTYPE	Char(8)	<p>This field contains the command type. Alphanumeric string whose meaning for commands is mission specific. For missions adhering to the Packet Telecommand Standard (see [RD-14]), this field will be used in order to identify the on-board control directives as follows:</p> <p>'R' – this is a Control Segment (a.k.a. Reset Map Command). The PCDF/CDF entries associated to this CCF entry define the structure of its frame data field (as opposed to standard commands for which the PCDF/CDF entries define the packet header and application data field).</p> <p>'F' - this is a Control Frame (a.k.a. BC frame i.e. Unlock and Set V(R)). The PCDF/CDF entries associated to this CCF entry define the structure of its frame data field (as opposed to standard commands for which the PCDF/CDF entries define the packet header and application data field).</p> <p>'S' - this is a Command Packet which does not contain the Packet Error Control Field (as opposed to standard commands to which the Releaser automatically appends the 16-bit Cyclic Redundancy Code (CRC) word, see [RD-6]). Typically, this option is used by EGSE systems for non-spacecraft commands.</p>	
5	CCF_CRITICAL	Char(1)	<p>Flag identifying the command 'criticality'.</p> <p>'Y' – if the command is critical (also referred to as hazardous).</p> <p>'N' – meaning non-critical.</p> <p>Note: the actual extra processing/protections required by critical commands is left up to mission specific implementation.</p>	'N'
6	CCF_PKTID	Char(8)	Name of the packet header (TCP_ID) to be used when constructing the command packet.	M
7	CCF_TYPE	Number(3)	<p>Service Type of the command.</p> <p>Integer value in the range (0....255).</p> <p>It can be Null if there is no PCDF entry of type 'T' in the definition of the associated packet header.</p>	
8	CCF_STYPE	Number(3)	<p>Service Sub-Type of the command.</p> <p>Integer value in the range (0....255).</p> <p>It can be Null if there is no PCDF entry of type 'S' in the definition of the associated packet header.</p>	
9	CCF_APID	Number(5)	<p>APID of the command.</p> <p>Integer value in the range (0....2047) for spacecraft commands and in the range (2048....65535) for non-spacecraft commands.</p> <p>It can be Null if there is no PCDF entry of type 'A' in the definition of the associated packet header.</p>	

Fi. Nr	Field name	Field Type	Description	Ma/Def
10	CCF_NPARS	Number(3)	Number of elements (i.e. parameters or fixed areas) associated with this command. Available for consistency checking against the total number of entries in the CDF table for this command.	
11	CCF_PLAN	Char(1)	Flag identifying the sources which are allowed to reference directly this command and use it in a planning file. 'A' – if the command is plannable by all sources. 'F' – if the command is plannable by flight dynamics only 'S' – if the command is plannable by the SOC only 'N' - if the command is not plannable by any source.	'N'
12	CCF_EXEC	Char(1)	Flag controlling whether this command can be loaded onto SCOS-2000 manual stacks stand-alone or only as part of command sequences. 'N' – if the command is to be treated cannot be loaded stand-alone. 'Y' – if no restriction applies.	'Y'
13	CCF_ILSCOPE	Char(1)	Flag identifying the default interlock associated to this command when loading it on a SCOS-2000 command source. 'G' - if the command will globally interlock the whole system, 'L' - if the command will interlock only the local commanding source, 'S' - if the command will interlock local commanding for the sub-system that the command belongs to, 'B' - if the command will interlock global commanding only for the sub-system that the command belongs to, 'N' - for no interlock.	'N'

Fi. Nr	Field name	Field Type	Description	Ma/Def
14	CCF_ILSTAGE	Char(1)	<p>The default type of verification stage to which an interlock associated to this command waits on:</p> <p>'R' – Reception/acceptance at the ground station 'U' – Uplink (also referred to as radiation/transmission) 'O' – On-board reception/acceptance (final uplink verification, also referred to as transfer) 'A' – Application acceptance 'C' – Completion</p> <p>Note that option 'O' is automatically handled as option 'U' when commanding in BD mode.</p> <p>This field is checked by the SCOS-2000 importer to avoid that a CEV Interlock Stage ('A' or 'C') is required for a command which does not foresee it in the CVP/CVS table. In case this would happen, CCF_ILSTAGE will be updated by the SCOS-2000 Importer by assigning the highest applicable level ('A' or 'O') for this command.</p>	'C'
15	CCF_SUBSYS	Number(3)	<p>Identifier of the subsystem which the command belongs to.</p> <p>Integer value in the range of (1....255).</p>	
16	CCF_HIPRI	Char(1)	<p>Flag identifying whether the command is implemented as high priority on-board i.e. its MAP ID is to be forced to a fixed value (see CCF_MAPID) and it cannot be execution time-tagged on-board (typically CPDU commands for a packet based mission).</p> <p>'Y' - if the command is high priority. 'N' - if it is not high priority.</p>	'N'
17	CCF_MAPID	Number(2)	<p>It contains the MAP ID value to be used for high priority commands.</p> <p>Integer value in the range (0...63).</p> <p>The field is irrelevant if the field CCF_HIPRI in not set to 'Y'.</p>	
18	CCF_DEFSET	Char(8)	<p>Name of the default parameter value set (PSV_PVSID) for the command.</p>	

Fi. Nr	Field name	Field Type	Description	Ma/Def
19	CCF_RAPID	Number(5)	<p>Redundant APID of the command. This field can be used in case of redundant on-board units supporting the same command but being allocated different APIDs. The SCOS-2000 TC SPACON application supports the ability to dynamically force the use of the redundant APID for a selected prime APID. In case this field is not Null, an entry showing the on-board sub-system mnemonic (as derived from CCF_SUBSYS) and the prime APID (CCF_APID) will be shown in the relevant option of the TC SPACON application. If this field is set, the prime APID (CCF_APID) must also be present.</p> <p>Integer value in the range (1...2047) (zero is not allowed) for spacecraft commands and in the range (2048...65535) for non-spacecraft commands.</p> <p>Note that all commands associated to the same prime APID must also be associated to the same redundant APID.</p>	
20	CCF_ACK	Number(2)	<p>Acknowledgement Flags of the command i.e. the integer value in decimal of the complete 'Ack' field of the Packet Data Field Header.</p> <p>Integer value in the range (0...15).</p> <p>It can be Null if there is no PCDF entry of type 'K' in the definition of the associated packet header.</p>	

3.3.3.2.1.1 Command routing: DST

This optional table defines the routing and destination of commands. Currently, it is only used by SCOS-2000 based EGSE systems. At most one record per commands Application ID (APID). Commands associated in the CCF table to an APID that is not present in this table are sent to a default destination i.e. "NCTRS".

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	DST_APID	Number(5)	<p>Command APID.</p> <p>Integer value in the range (0...65535).</p>	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
2	DST_ROUTE	Char(30)	<p>Alphanumeric string describing the routing and destination (referred to as “Destination ID”) to be used for all commands (CCF_APID) associated to the APID specified in DST_APID. The Destination ID is a string made of up to 3 fields separated by ‘.’ (dot).</p> <p>The first field represents the name of the system to which the SCOS-2000 Releaser will deliver the command. Possible values are:</p> <p>‘NCTRS’ – This option can only be used for spacecraft commands. This is the default for all APIDs not specified in this table.</p> <p>‘EGSE’ – This option can be used for both spacecraft and non-spacecraft commands.</p> <p>The second and third fields are only applicable to commands not delivered to the NCTRS (first field set to ‘EGSE’). The second field represents the type of request and its layout. Currently SCOS-2000 supports the following two types of requests:</p> <p>‘SCOE’ – for non-spacecraft commands.</p> <p>‘TC’ – for spacecraft commands.</p> <p>The third field is only applicable to spacecraft commands not routed via the NCTRS. It represents the address of the ground equipment that will issue the command to the spacecraft.</p> <p>Valid examples for the Destination ID string are:</p> <p>‘NCTRS’ (spacecraft command routed via the NCTRS)</p> <p>‘EGSE.SCOE’ (non-spacecraft command)</p> <p>‘EGSE.TC.0x8800’ (spacecraft command routed via the SCOE and TC front-end 0x8800.</p>	M

3.3.3.2.2 Command parameters: cpc

This table defines the characteristics of command parameters. One record per command parameter. Prior to describing the details of the CPC table fields, it is necessary to describe the handling of command parameter default values within SCOS-2000.

Command parameter default values

A command parameter may be associated to a default value at each level of instantiation i.e. at the level of parameter definition itself (CPC_DEFVAL), at the level of command definition (i.e. when defining the structure of a command containing the parameter, CDF_VALUE) or at the level of sequences definition (i.e. when using any command containing the parameter as a sequence element, SDF_VALUE). The same parameter can be re-used in several command definitions which on their turn can be re-used in several sequences definition. In order to enable a centralised maintenance of command parameter default values (whenever required), SCOS-2000 supports the ability to propagate default values from the lower to the higher levels of instantiation. This is achieved by providing the ability to refer to the default value specified at the previous instantiation level (option ‘D’ for fields CDF_INTER and SDF_VTYPE). Further, it is possible to ‘fix’ the value of a parameter at any instantiation level i.e. it is possible

to prohibit editing of the command parameter default value in the context of command definition (CDF_ELTYPE='F') or in the context of sequence definition (SDF_FTYPE='F').

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CPC_PNAME	Char(8)	Unique name of the command parameter.	M
2	CPC_DESCR	Char(24)	Textual description of the command parameter.	
3	CPC_PTC	Number(2)	Parameter Type Code. It identifies the parameter type i.e. unsigned integer, integer, real, octet string, character string, absolute time, relative time. Parameter Type Codes supported by SCOS-2000 and their interpretation are specified in Appendix A. Appendix A also specifies the format in which the raw values have to be specified for each supported parameter type.	M
4	CPC_PFC	Number(5)	Parameter Format Code. Along with the Parameter Type Code (CPC_PTC) this field identifies the parameter length (see Appendix A). Integer value in a range compatible with the specified PTC as specified in Appendix A.	M
5	CPC_DISPFMT	Char(1)	Flag controlling the input and display format of the engineering values for calibrated parameters and for time parameters. 'A' – ASCII (if CPC_CATEG='T'). 'I' – signed integer (if CPC_CATEG='C' and CCA_ENGFMT='I') 'U' – unsigned integer (if CPC_CATEG='C' and CCA_ENGFMT='U') 'R' – real (if CPC_CATEG='C' and CCA_ENGFMT='R') 'T' – absolute time (if CPC_PTC=9) 'D' – delta/relative time (if CPC_PTC=10). The formats in which values are to be input/displayed for the different representation types are specified in Appendix A.	'R'
6	CPC_RADIX	Char(1)	This is the default radix that the raw representation of the parameter will be displayed in if it is unsigned integer. 'D' – Decimal 'H' – Hexadecimal 'O' – Octal Note, this maybe over-ridden by a mission configurable set of allowed radices.	'D'
7	CPC_UNIT	Char(4)	Engineering unit mnemonic e.g. 'VOLT' to be displayed besides command parameter values.	

Fi. Nr	Field name	Field Type	Description	Ma/Def
8	CPC_CATEG	Char(1)	<p>Flag identifying the type of calibration associated to the parameter.</p> <p>'C' – numeric calibration, therefore CPC_CCAREF shall not be Null.</p> <p>'T' – textual calibration, therefore CPC_PARREF shall not be Null.</p> <p>'B' – both calibrations (not supported yet by SCOS-2000).</p> <p>'N' – implies no calibration.</p> <p>This field must be left Null or set to 'N' for command parameters whose rawrepresentation is not of a numeric nature (i.e. strings, times).</p>	'N'
9	CPC_PRFREF	Number(4)	<p>This field contains the ID of an existing parameter range set (PRF_NUMBR) to which the parameter will be associated. The ranges defined within the set will determine the possible values for the parameter. The format of the selected range set (PRF_DSPFMT) must be compatible with the type of the command parameter (described by CPC_PTC or CPC_DISPFMT depending on the used representation i.e. PRF_INTER).</p>	
10	CPC_CCAREF	Number(4)	<p>This field contains the ID of an existing numeric calibration curve set (CCA_NUMBR) to which the parameter will be associated. The value formats of the selected curve set (CCA_RAWFMT and CCA_ENGFMT) must be compatible with the type of the command parameter (described by CPC_PTC) and its engineering format (CPC_DISPFMT).</p> <p>It must be Null if CPC_CATEG is Null, 'N' or 'T'. It cannot be Null if CPC_CATEG is 'C' or 'B'.</p>	
11	CPC_PAFREF	Number(4)	<p>This field contains the ID of an existing textual calibration set (PAF_NUMBR) to which the parameter will be associated. The aliases contained within the set may be used as default values for the parameter. The selected alias set (PAF_RAWFMT) must be compatible with the type of the command parameter (described by CPC_PTC).</p> <p>It must be Null if CPC_CATEG is Null, 'N' or 'C'. It cannot be Null if CPC_CATEG is 'T' or 'B'.</p>	
12	CPC_INTER	Char(1)	<p>Flag identifying the representation used to express the parameter default value.</p> <p>'R' – if the default value (CPC_DEFVAL) is expressed using a raw representation.</p> <p>'E' - if the default value (CPC_DEFVAL) is expressed using an engineering representation. This option can only be used for parameters which are associated to a calibration curve (i.e. CPC_CATEG = 'T' or 'C').</p>	'R'

Fi. Nr	Field name	Field Type	Description	Ma/Def
13	CPC_DEFVAL	Char(17)	<p>This field contains the default value for this parameter. It is used if specified so in an associated CDF entry (CDF_INTER = 'D'). Associating a default value to a command parameter (as opposed to associating the default value in the command definition) is useful if the same default value has to be used for all instances of a parameter in all command definitions.</p> <p>This parameter default value must be expressed using the appropriate representation as specified in CPC_INTER. Raw values must be expressed in a format compatible with the type of the parameter (as specified by CPC_PTC and CPC_RADIX for unsigned integer). Engineering values must be expressed in a format compatible with the engineering representation of the parameter (CPC_DISPFMT).</p>	

3.3.3.2.3 Commands definition: cdf

This table defines the structure of the commands in terms of constituent elements. One record per command element (i.e. parameter or fixed area).

Prior to describing the details of the CDF table fields, it is necessary to introduce two new concepts (repeated groups and dynamic defaults).

Repeated Groups

In order to satisfy all the possible command structures which are supported by PUS, the concept of repeated groups needs to be addressed. This is where certain parameters in the command may be repeated a variable number of times.

If a group of consecutive parameters may be repeated more than once, then a counter field parameter (typically named N or N1 etc.) must be present in the command before the start of the group. If this counter parameter can be identified and the size of the repeated group given, then the structure of the command can be reproduced.

Below is an example of a command with a repeated group. The **Distribute On/Off Command** (2,1) is simply an *address* parameter which can be repeated N times. This can be modelled by the following pieces of information:

There are two parameters, one at position 0 and one at position 1. The first parameter is a counter whose group size is 1.

At run time, the user may be asked to overwrite the default value of N already set in the database (if it has not been fixed), and then the *address* parameter will be repeated contiguously using that value.

N	ADDRESS
Unsigned Integer	Unsigned Integer

< Repeated N Times >

This method will also work with command structures that have nested groups as the example below shows. The **Disable Transmission of Telemetry Source Packets Request** (14,2) command has two repeated groups. The *sub-type* parameter is repeated N2 times. However, this repeated group plus the *type* are themselves in a group repeated N1 times.

N1	TYPE	N2	SUB-TYPE
----	------	----	----------

Unsigned Integer	Enumerated	Unsigned Integer	Unsigned Integer
------------------	------------	------------------	------------------

< Repeated N2 Times>

<----- Repeated N1 Times----->

To model this command we require the same information as for the previous example. I.e. the position of each parameter relative to the start of the command (unexpanded) plus for each parameter that is a counter, we also need the size of the group.

- N1 is at position 0 with a group size of 3 (*type*, N2, *sub-type*)
- *type* is at position 1 (e.g. bit offset 8) with a group size of 0 (it is not a counter)
- N2 is at position 2 (e.g. bit offset 16) with a group size of 1 (*sub-type*)
- *sub-type* is at position 3 (e.g. bit offset 24) with a group size of 0 (it is also not a counter).

When this command is loaded, it will be expanded from left to right. Firstly, given that the counter parameters have not been set as fixed, the user will be asked for the value of N1. This will then generate the next three parameters N1 times.

Then, for each instance of the N2 counter parameter (remember there will now be N1 of them) the user is asked for its value, and each instance of the *sub-type* will be repeated N2 times.

The only additional information that needs to be stored in the database, is whether or not an instance of a command parameter (in the CDF) is a counter or not. If so, how big is the group that it belongs to. This is modelled by a single additional field, which if set to greater than zero indicates a counter. Its value gives the group size.

Of course, it is still possible for a user to define a fixed command (at database definition time) for every combination of parameters required. In the first example, if the user required a command with two *address* parameters, then the N counter parameter would be defined as fixed (at value 2), and two further parameters would be defined explicitly. The group size of N would be 0, as this would not be treated as a repeatable group. Then at command load-time, there would be no prompt for N's value and no further parameter generation.

Finally, the user could set the N counter to fixed, but still set the group size to be greater than zero, which will force an expansion of the command without user prompting. This saves the user from having to define unnecessary entries in the CDF if he knows certain parameters need to be repeated.

Dynamic Defaults

Each editable command element may be associated to a static default value. This value can be overridden by the user when loading the command on a SCOS-2000 manual stack or when inserting the command in a sequence definition. In addition, SCOS-2000 provides the ability to override the static default with a 'dynamic default' value extracted from a monitoring parameter. This is useful in all cases where a portion of the command structure has to be initialised based on the current state on-board. It is also useful in the case that the value of command parameters used across many commands (e.g. to control their routing on-board) has to be controlled centrally (in this case one could associate the command parameter to a dynamic default linked to a User Defined Constant and force the required value of the User Defined Constant via TM SPACON).

Dynamic defaults provide functionality also supported in previous systems (e.g. Spacecraft Register Images) but in a more generic manner. Note that command parameters associated to a dynamic default must also be associated to a static default value (expressed in raw representation) In fact, the static default value is required in order to allow the command to be loaded on a command source (the TM values are used as a default only after command loading). In case an explicit value is specified on-line for a command parameter associated to the dynamic default, the extraction of the parameter value from the telemetry is stopped.

Fi. Nr	Field name	Field Type	Description	Ma/Def
--------	------------	------------	-------------	--------

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CDF_CNAME	Char(8)	The name of the command (CCF_CNAME) to which an element is associated.	M
2	CDF_ELTYPE	Char(1)	Type of the command element: 'A' – for a fixed Area, 'F' – for a parameter which has been Fixed and is non-editable or 'E' – for a parameter that is Editable.	M
3	CDF_DESCR	Char(24)	Textual description used for fixed areas only (CDF_ELTYPE='A').	
4	CDF_ELLEN	Number(4)	This field specifies the length of the command element expressed in number of bits. For fixed areas (CDF_ELTYPE = 'A'), this is actually used by SCOS-2000 to determine the element length in order to encode the command. For (fixed or editable) command parameters this field is not used by SCOS-2000 as the element length is derived from the PTC/PFC pair of the referenced parameter (CDF_PNAME). However, it is expected that this field is properly populated with the parameter length in order to be able to check the consistency of the commands and parameters definition. If the element is a variable length parameter (e.g. PTC=7, PFC = 0), then this value must be set to zero.	M
5 *	CDF_BIT	Number(4)	This field specifies the bit offset of this element relative to the start of the unexpanded command application data field (i.e. the end of the packet header, including the data field header where applicable, as specified in the PCDF table). The bit offset of command elements following a repeated group shall be calculated assuming no repetition (i.e. counters value = 1). Unsigned integer number of bits.	M
6	CDF_GRPsize	Number(2)	If this element is a group repeater parameter, then this field's value is the number of following command elements belonging to the group to be repeated 'N' times at load-time. Otherwise its value has to be left Null or set to zero.	'0'
7	CDF_PNAME	Char(8)	This field contains the name of a command parameter which is located in the command (CPC_PNAME). A parameter must be entered when the user has defined the CDF_ELTYPE to be 'F' or 'E'. This field is irrelevant for fixed areas (CDF_ELTYPE='A').	

Fi. Nr	Field name	Field Type	Description	Ma/Def
8	CDF_INTER	Char(1)	<p>Flag identifying the representation in which the (fixed or default) value (CDF_VALUE) is expressed for this command element.</p> <p>'R' - if the value is expressed using a raw representation.</p> <p>'E' - if the value is expressed using an engineering representation. This option can only be used for parameters which are associated to a calibration curve (i.e. CPC_CATEG = 'T' or 'C').</p> <p>'D' - if the value is to be taken from the corresponding CPC entry (CPC_DEFVAL),</p> <p>'T' - if the default value is to be taken from the value of the monitoring parameter given by CDF_TMID (dynamic default). This option can only be used for editable parameters (CDF_ELTYPE='E'). Note that a static default value for this parameter must be specified in the CDF_VALUE field using a raw representation.</p> <p>This field is irrelevant for fixed areas (CDF_ELTYPE='A').</p>	'R'
9	CDF_VALUE	Char(17)	<p>This field specifies the default or fixed value for the given element. If the element type is a fixed area or a group repeater then this value must be a raw unsigned integer value expressed in Hex. If the command element is a (fixed or editable) parameter, this value must be expressed using the appropriate representation i.e. raw unless option 'E' is used in CDF_INTER. Raw values must be expressed in a format compatible with the type of the parameter (as specified by CPC_PTC and CPC_RADIX for unsigned integer). Engineering values must be expressed in a format compatible with the engineering representation of the parameter (CPC_DISPFMT).</p> <p>This field cannot be left Null if the command parameter is fixed (CDF_ELTYPE = 'F') or is associated to a dynamic default (CDF_INTER = 'T'). If the parameter is editable (CDF_ELTYPE = 'E') then this field maybe left Null indicating that the user MUST enter a value before loading the command.</p> <p>This field is irrelevant if option 'D' is used in CDF_INTER.</p>	
10	CDF_TMID	Char(8)	<p>This field contains the name of a monitoring parameter used to extract dynamic default values. This field is only applicable if option 'T' is used in CDF_INTER. The PTC/PFC values of the selected monitoring parameter must be the same as the PTC/PFC values of the corresponding command parameter given by CDF_PNAME.</p>	

3.3.3.2.4 Commands pre-transmission validation: ptv

This table describes which expressions are used by which commands to perform static PTV. One record per validation check contributing to the static PTV of a command. Multiple checks defined for a command are implicitly ANDed together and therefore only one instance of a PTV parameter should be defined per command.

Remark: the validity of the monitoring parameters used for command validation is implicitly taken into account by SCOS-2000 i.e. a PTV check is only passed if the validation parameter is valid and its value is as required.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PTV_CNAME	Char(8)	Name of the command (CCF_CNAME) using this validation expression.	M
2 *	PTV_PARNAM	Char(8)	Name of an existing monitoring parameter (PCF_NAME) to be used for the validation check. Any type of monitoring parameters can be used for validation i.e. TM parameters or synthetic parameters. However, only monitoring parameters whose raw representation is of numerical nature can be used (i.e. no strings, times).	M
3	PTV_INTER	Char(1)	Flag identifying the representation selected to specify the validation value (PTV_VAL) to be used for the verification check. 'R' - if the value is expressed using a raw representation. Note that in case of synthetic parameters the raw representation is the source one i.e. the one resulting from the associated OL or hard-coded definition. 'E' - if the value is expressed using an engineering representation (i.e. calibrated value). This option can only be used for validation parameters which are associated to a calibration curve (i.e. PCF_CURTX is not Null).	'R'
4	PTV_VAL	Char(17)	This field contains the value that the validation parameter (PTV_PARNAM) must equal in order to pass the PTV check. This value must be expressed using the appropriate representation i.e. raw unless option 'E' is used in PTV_INTER. Raw values must be expressed in decimal in a format compatible with the type of the validation parameter (as specified by PCF_PTC). Engineering values must be expressed in a format compatible with the engineering representation of the validation parameter (i.e. ASCII if PCF_CATEG = 'S' or as specified in the associated CAF_ENGFMT for numerically calibrated parameters).	M

3.3.3.3 Command Sequences

The following tables define the contents of command sequences, which are ordered lists of elements which can be commands or sequences themselves. SCOS-2000 supports the definition of sequences hierarchies with any level of nesting. Basically, a nested sequence is handled as a 'macro-command' which is expanded at sequence loading time. The only constraint is that a sequence should never contain itself! Each sequence element (command or sequence) has associated with it parameters which are given implicitly or explicitly a value as part of the sequence definition. Also, the sequence itself may be associated with parameters (referred to as 'formal parameters'), which are

effectively place holders within the sequence that can be used to parameterise the behaviour of the sequence. A formal parameter can either replace an element (requiring the user to provide a command or sequence ID), or it can replace an element parameter (which can be a command parameter, or if the element is a sequence, then it will be a parameter of that nested sequence).

Remark: in order to describe the concepts of parameters associated to sequences containing other parametric sequences, it is important to distinguish in terminology between the parameters of the sequence being used as a sequence element (i.e. a nested sequence) and the parameters of the sequence being defined and containing the nested sequence. A formal parameter refers to an argument of the current sequence. A sequence parameter refers to the parameter of a nested sequence. When describing the nested sequence, the sequence parameters are then (and only then) referred to as formal parameters.

The CSF table describes the top level of the sequence hierarchy, i.e. it defines the sequence itself.

The CSS table contains an entry for each element in the sequence. An element can either be a sequence or command, or it can be a formal parameter of the current sequence, allowing the user to provide at a later stage the command or the sequence name to be used as an element.

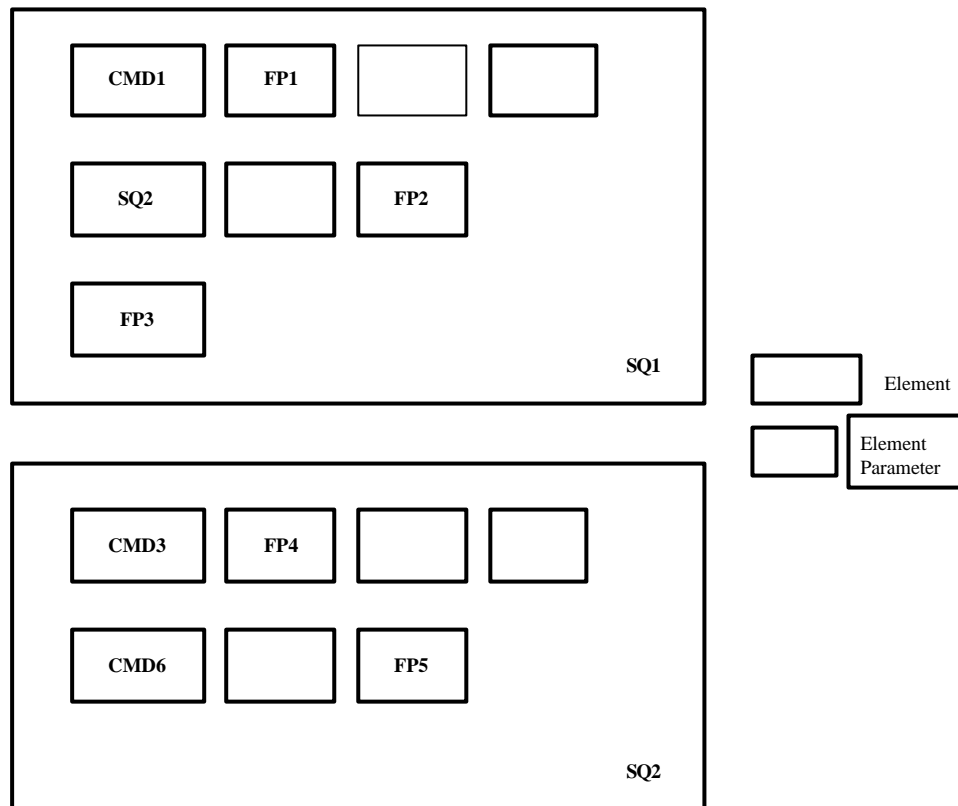
The SDF table defines the values of the editable element parameters. An element parameter can be:

- a command parameter, whose value is explicitly defined in the SDF entry or can be propagated from the original command definition default value in the CDF.
- a sequence parameter, whose value is explicitly defined in the SDF entry or can be propagated from the default value in the CSP for the nested sequence.
- a command or a sequence parameter which are not given any value as part of the sequence definition but are mapped to a formal parameter (whose default value is defined in the CSP associated with this sequence). It is possible to associate several element parameters to the same formal parameter (provided that they are all of the same type i.e. that they share the same raw and engineering value formats).

The CSP defines the formal parameters associated with the current sequence. Each entry can either be a command/sequence name, or it can be associated to one or more element (command/sequence) parameters. Formal parameters default values are defined in the CSP entry.

Remark: the first SCOS-2000 release will not support formal parameter of type command or sequence name.

The following diagram shows an example of all cases.



SQ1 is a sequence containing 3 elements:

- command element CMD1 with 3 command parameters
- sequence element SQ2 with 2 sequence parameters
- 1 formal parameter element.

SQ2 is a sequence containing 2 elements:

- command CMD3 with 3 command parameters
- command CMD6 with 2 command parameters

SQ1 has 3 formal parameters - FP1, FP2 and FP3 (FP1 is a command parameter, FP2 is a sequence parameter and FP3 is a command/sequence name).

SQ2 has 2 formal parameters - FP4 and FP5 (both are command parameters).

From the point of view of SQ1, SQ2 has 2 sequence parameters of which the second is a formal parameter of SQ1.

Seen from SQ2, SQ2 has 2 formal parameters (FP4 and FP5).

Remark: Although in this example unique formal parameter names have been used across the sequences, there is no restriction on this. Formal parameter names must be unique only within the same sequence. Therefore, FP2 could have been called FP5.

The tables will contain:

2 entries in the CSF - one for each sequence SQ1 and SQ2.

5 entries in the CSS - 3 for the elements of SQ1 and 2 for the elements of SQ2.

10 entries in the SDF - 5 for each sequence.

5 entries in the CSP - 3 for SQ1 (FP1, FP2, FP3) and 2 for SQ1 (FP4 and FP5).

At load time when the user specifies a value for FP2, that value will be propagated and will become the value for FP5, which in turn will become the value of the second parameter of CMD6.

Remark: the element in sequence SQ1 defined by the formal parameter FP3 has no parameters itself as it is not known what the element will be. Therefore, its parameters are taken from the defaults specified by the command or sequence once its name is known. It is thus not possible to use commands/sequences which are not fully defined i.e. for which parameters are left without any default value.

3.3.3.1 Command sequence characteristics: csf

This table describes the command sequences. One record per command sequence.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CSF_NAME	Char(8)	Unique name of the command sequence.	M
2	CSF_DESC	Char(24)	Textual description of the command sequence.	M
3	CSF_DESC2	Char(64)	Detailed description of the command sequence.	
4	CSF_IFFT	Char(1)	Flag identifying whether or not the sequence contains execution time-tagged commands. 'Y' - if the sequence or any of its nested sequences contain execution time tagged commands. The user will be prompted at load time to provide an absolute time which will be used as a reference to calculate all the individual execution time-tagged commands. 'B' - if the commands (directly or indirectly) contained within the sequence are all execution time tagged commands. In this case, the user will have at load time the option to load the sequence as on-board time-tagged (in which case only the commands execution time-tag will be considered) or as ground time-tagged (in which case only the command release time-tags will be considered and all commands will be released for immediate execution on-board) 'N' - if the sequence and its nested sequences do not contain any execution time tagged command.	'N'
5	CSF_NFPARS	Number(3)	Number of formal parameters of this sequence. Available for consistency checking against the number of records associated to this sequence in the CSP table.	
6	CSF_ELEMS	Number(5)	Number of elements in this sequence. Available for consistency checking against the number of records associated to this sequence in the CSS table.	

Fi. Nr	Field name	Field Type	Description	Ma/Def
7	CSF_CRITICAL	Char(1)	Flag identifying the sequence 'criticality'. 'Y' – if the sequence is critical (also referred to as hazardous). 'N' – meaning non-critical. Note: the actual extra processing/protections required by critical sequences is left up to mission specific implementation.	'N'
8	CSF_PLAN	Char(1)	Flag identifying the sources which are allowed to reference directly this sequence and use it in a planning file. 'A' – if the sequence is plannable by all sources. 'F' – if the sequence is plannable by flight dynamics only 'S' – if the sequence is plannable by the SOC only 'N' – if the sequence is not plannable by any source.	'N'
9	CSF_EXEC	Char(1)	Flag controlling whether this sequence can be loaded onto SCOS - 2000 manual stacks stand-alone (or only as a nested sequence i.e. as an element of another command sequence). 'N' – if the sequence is to be treated cannot be loaded stand-alone. 'Y' – if no restriction applies.	'Y'
10	CSF_SUBSYS	Number(3)	Identifier of the subsystem which the sequence belongs to. Integer value in the range of (1...255).	
11	CSF_GENTIME	Char(17)	Time when sequence was generated.	
12	CSF_DOCNAME	Char(32)	Document name from which it was generated.	
13	CSF_ISSUE	Char(10)	Issue number of document described above.	
14	CSF_DATE	Char(17)	Issue date of document described above.	
15	CSF_DEFSET	Char(8)	Name of the default parameter value set (PSV_PVSID) for the sequence.	

3.3.3.2 Command sequences definition: css

This table defines the sequence elements in relation to the command sequence they belong to. One record per command sequence element (command or command sequence) of a command sequence.

Remark: all attributes associated to commands as part of the sequence definition should be considered as defaults in that the command execution requests can be edited in a manual stack after loading the sequence.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CSS_SQNAME	Char(8)	Name of the sequence which the element belongs to.	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
2	CSS_COMM	Char(32)	Additional comment to be associated with the sequence element. If the element is of type comment (CSS_TYPE= 'T'), then this text is displayed in SCOS-2000 command sources.	
3 *	CSS_ENTRY	Number(5)	Entry number for the sequence element. It gives the order in which the element resides in the sequence. Sequences cannot contain gaps, therefore consecutive entries must be specified.	M
4	CSS_TYPE	Char(1)	Flag identifying the element type. It may take one of the following five values: 'C' – Command 'S' – Sequence 'T' - Textual comment - non-uplinkable entry (not supported by SCOS-2000) 'F' - Formal parameter representing a command (future extension) 'P' - Formal parameter representing a sequence (future extension)	M
5	CSS_ELEMID	Char(8)	Element (identifier) for the current entry within the sequence. If element type (CSS_TYPE) is 'C', it must be a valid command name (CCF_CNAME). If element type (CSS_TYPE) is 'S' it must be a valid sequence name (CSF_SQNAME). If element type (CSS_TYPE) is 'F' or 'P' it must be a valid formal parameter name (CSP_FPNAME). If element type (CSS_TYPE) is 'T' it must be Null.	
6	CSS_NPARS	Number(3)	Number of editable parameters this element has (i.e. no.of entries in the SDF for this element id). This field is only relevant if the element type (CSS_TYPE) is 'C' or 'S'.	
7	CSS_MANDISP	Char(1)	Flag controlling whether the command requires explicit manual dispatch (ARM&GO) in a manual stack 'Y' - if this command requires separate manual dispatch. 'N' - if this element is dispatched together with the previous command. This field is irrelevant if the element is not a command.	'N'



Fi. Nr	Field name	Field Type	Description	Ma/Def
8	CSS_RELTYPE	Char(1)	Flag controlling how the field CSS_RELTIME has to be used to calculate the command release time -tag or the nested sequence release start time (see description of CSS_RELTIME below) 'R' - if this element's release/start time has to be calculated relative to the release time of the previous command in the expanded sequence (or relative to its own dispatch time in case this is a command associated to a manual dispatch flag). 'A' - <i>if the element's release/start time has to be calculated relative to the start time of the sequence that this element belongs to (CSS_SQNAME). For the top-level sequence, the start time must be provided by the user at load-time.</i>	'R'

Fi. Nr	Field name	Field Type	Description	Ma/Def
9	CSS_RELTIME	Char(8)	<p>This field contains the delta release time -tag for the sequence element. It is used as follows:</p> <ul style="list-style-type: none"> ➤ If this element is a command (CSS_TYPE = 'C') and CSS_RELTYPE='R' and CSS_MANDISP='Y', then the absolute release time of this command is calculated by adding this field (CSS_RELTIME) to the command dispatch time. ➤ If this element is a command (CSS_TYPE = 'C') and CSS_RELTYPE='R' and CSS_MANDISP='N', then the absolute release time of this command is calculated by adding this field (CSS_RELTIME) to the release time of the last command prior to this command in the expanded top level parent sequence (i.e. in case the previous element in this sequence is a nested sequence, the absolute time is calculated with respect to the last command of the previous element, which will become the previous command once the sequence is expanded on a command source). ➤ <i>If this element is a command (CSS_TYPE = 'C') and CSS_RELTYPE='A', then the absolute release time of this command is calculated by adding this field (CSS_RELTIME) to the release start time of the current sequence (CSS_SQNAME).</i> ➤ If this element is a sequence (CSS_TYPE = 'S') and CSS_RELTYPE='R', then the release start time of this nested sequence is calculated by adding this field (CSS_RELTIME) to the release time of the last command in the expanded parent sequence prior to this nested sequence (i.e. in case the previous element in this sequence is a nested sequence, the absolute time is calculated with respect to the last command of the previous element, which will become the previous command once the sequence is expanded on a command source). ➤ <i>If this element is a sequence (CSS_TYPE = 'S') and CSS_RELTYPE='A', then the release start time of this nested sequence is calculated by adding this field (CSS_RELTIME) to the release start time of the current sequence (CSS_SQNAME).</i> ➤ If this field is Null and this element is a command (CSS_TYPE= 'C'), it implies that this command is not release time -tagged (ASAP release condition). If this field is Null and this element is a sequence (CSS_TYPE= 'S'), it implies that this sequence has zero delta release start time. <p>The delta release time -tag is a time interval to be specified in the format: hh.mm.ss (up to 23.59.59).</p>	

Fi. Nr	Field name	Field Type	Description	Ma/Def
10	CSS_EXETIME	Char(16)	<p>This field contains the delta execution time -tag for the sequence element. It is used as follows:</p> <ul style="list-style-type: none"> ➤ If this element is a command (CSS_TYPE = 'C') and CSS_PREVREL='R', then the absolute execution time of this command is calculated by adding this field (CSS_EXETIME) to the execution time of the last execution time-tagged command prior to this command in the expanded top level parent sequence (i.e. in case the previous element in this sequence is a nested sequence, the absolute time is calculated with respect to the last execution time -tagged command of the previous element, which will become the previous execution time-tagged command once the sequence is expanded on a command source). ➤ If this element is a command (CSS_TYPE = 'C') and CSS_PREVREL='A', then the absolute release time of this command is calculated by adding this field (CSS_EXETIME) to the execution start time of the current sequence (CSS_SQNAME). ➤ If this element is a sequence (CSS_TYPE = 'S') and CSS_PREVREL='R', then the execution start time of this nested sequence is calculated by adding this field (CSS_EXETIME) to the execution time of the last execution time -tagged command prior to this command in the expanded top level parent sequence (i.e. in case the previous element in this sequence is a nested sequence, the absolute time is calculated with respect to the last execution time-tagged command of the previous element, which will become the previous execution time -tagged command once the sequence is expanded on a command source). ➤ If this element is a sequence (CSS_TYPE = 'S') and CSS_PREVREL='A', then the execution start time of this nested sequence is calculated by adding this field (CSS_EXETIME) to the execution start time of the current sequence (CSS_SQNAME). ➤ If this field is Null and this element is a command (CSS_TYPE= 'C'), it implies that this command is not execution time-tagged (i.e. immediately executed on-board as soon as received). If this field is Null and this element is a sequence (CSS_TYPE= 'S'), it implies that this sequence has zero delta execution start time. <p>The delta execution time -tag is a time interval to be specified in the format: [+][ddd.]hh.mm.ss[.mmm]</p>	

Fi. Nr	Field name	Field Type	Description	Ma/Def
11	CSS_PREVREL	Char(1)	<p>Flag controlling how the field CSS_EXTIME has to be used to calculate the command execution time -tag or the nested sequence execution start time (see description of field CSS_EXTIME above).</p> <p>'R' - if this element's execution/start time has to be calculated relative to the execution time-tag of the previous time-tagged command in the expanded sequence.</p> <p>'A' - if this element's execution/start time has to be calculated relative to the start time of the sequence that this element belongs to (CSS_SQNAME). For the top-level sequence, the start time is provided by the user at load-time.</p>	'R'
12	CSS_GROUP	Char(1)	<p>This field defines the grouping condition for the sequence element and (only applicable for commands). This controls whether this command and the following/previous commands are to be released as a batch of commands (i.e. they cannot be interleaved with other commands being released by other command sources).</p> <p>'S' – Start of the group</p> <p>'M'- Middle of a group</p> <p>'E' – End of the group</p> <p>Each command group shall contain at least one command associated to the flag 'S' and one associated to the flag 'E' (with a configurable maximum number of grouped commands associated to the flag 'M' in between).</p> <p>This field can be Null if the command within the sequence is not grouped with others. This field is irrelevant for sequence elements which are not commands.</p>	

Fi. Nr	Field name	Field Type	Description	Ma/Def
13	CSS_BLOCK	Char(1)	<p>This field defines the blocking condition for the sequence element (only applicable for commands). This controls whether this command and the following/previous commands are to be released in a single data unit (this is achieved by aggregating all packets into a single segment data field).</p> <p>'S' – Start of the block 'M'- Middle of a block 'E' – End of the block</p> <p>Each command block shall contain at least one command associated to the flag 'S' and one associated to the flag 'E' (with a configurable maximum number of blocked commands associated to the flag 'M' in between). It is possible to include a complete block within a group but not a group within a block.</p> <p>Due to the fact that blocked commands are aggregated into a single segment, it shall be ensured that the total length of the command block does not exceed a mission configurable limit. Further, it is not allowed to include high-priority commands (CCF_HIPRI='Y') in a block.</p> <p>This field can be Null if the command within the sequence is not blocked with others. This field is irrelevant for sequence elements which are not commands.</p>	

Fi. Nr	Field name	Field Type	Description	Ma/Def
14	CSS_ILSCOPE	Char(1)	<p>Flag identifying the type of the interlock associated to this sequence element (only applicable for commands).</p> <p>'G' – if the command will globally interlock the whole system, 'L' – if the command will interlock only the local commanding source, 'S' – if the command will interlock local commanding for the sub-system that the command belongs to, 'B' – if the command will interlock global commanding only for the sub-system that the command belongs to, 'F' – if the command will interlock only the local commanding source and only in case of verification failure (i.e. no WAIT state), 'T' – if the command will interlock local commanding for the sub-system that the command belongs to and only in case of verification failure (i.e. no WAIT state), 'N' – for no interlock.</p> <p>In the case of commands belonging to a Block (CSS_BLOCK not Null) or a Group (CSS_GROUP not Null), options 'G' and 'L' are only allowed for the last command in the Block/Group (i.e. CSS_BLOCK/CSS_GROUP = 'E').</p> <p>If this field is left Null, then the interlock information is taken from the command definition (CCF_ILSCOPE). This field is irrelevant for sequence elements which are not commands.</p>	

Fi. Nr	Field name	Field Type	Description	Ma/Def
15	CSS_ILSTAGE	Char(1)	<p>The type of verification stage to which an interlock associated to this sequence element (only applicable for commands) waits on:</p> <p>‘R’ – Reception/acceptance at the ground station ‘U’ – Uplink (also referred to as radiation/transmission) ‘O’ – On-board reception/acceptance (final uplink verification, also referred to as transfer) ‘A’ – Application acceptance ‘C’ – Completion</p> <p>Note that option ‘O’ is automatically handled as option ‘U’ when commanding in BD mode.</p> <p>This field is checked by the SCOS-2000 importer to avoid that a CEV Interlock Stage (‘A’ or ‘C’) is required for a command which does not foresee it in the CVP/CVS table. In case this would happen, CCF_ILSTAGE will be updated by the SCOS-2000 Importer by assigning the highest applicable level (‘A’ or ‘O’) for this command.</p> <p>If this field is left Null, then the interlock information is taken from the command definition (CCF_ILSTAGE). This field is irrelevant for sequence elements which are not commands.</p>	
16	CSS_DYNPTV	Char(1)	<p>Flag controlling whether the dynamic PTV checks (i.e. feasibility checks, time checks, interlocks) shall be overridden for this sequence element (only applicable for commands).</p> <p>‘Y’ – dynamic PTV is to be overridden for this command. ‘N’ – dynamic PTV is not to be overridden for this command.</p> <p>This field is irrelevant if the sequence element is not a command.</p>	‘N’
17	CSS_STAPTV	Char(1)	<p>Flag controlling whether the static (TM based) PTV check shall be overridden for this sequence element (only applicable for commands).</p> <p>‘Y’ – static PTV is to be overridden for this command. ‘N’ – static PTV is not to be overridden for this command.</p> <p>This field is irrelevant if the sequence element is not a command.</p>	‘N’
18	CSS_CEV	Char(1)	<p>Flag controlling whether the CEV checks shall be disabled for this sequence element (only applicable for commands).</p> <p>‘Y’ – CEV checks are to be disabled for this command. ‘N’ – CEV checks are not to be disabled for this command.</p> <p>This field is irrelevant if the sequence element is not a command.</p>	‘N’

3.3.3.3 Command sequence element parameters: sdf

This table specifies how to allocate a value to the editable parameters of sequence elements. One record per editable parameter of sequence elements. Note that for each editable element parameter the 'source' of the value to be used when loading the sequence has to be identified. This can be a database field (either specified in this table or in other tables) or a reference to a formal parameter from which the value is inherited.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	SDF_SQNAME	Char(8)	Name of the sequence containing the element to which this parameter belongs (CSF_SQNAME).	M
2 *	SDF_ENTRY	Number(5)	Entry number of the sequence element to which this parameter belongs (CSS_ENTRY). It must point to a sequence element of type command or sequence (CSS_TYPE='C' or 'S').	M
3	SDF_ELEMID	Char(8)	Name of the sequence element to which this parameter belongs (CSS_ELEMID).	M
4 *	SDF_POS	Number(4)	If the sequence element is a command, then this field specifies the bit offset of the parameter in the command (i.e. CDF_BIT). If the element is a nested sequence, then this field specified the sequence parameter position (i.e. the CSP_FPNUM of the formal parameter for that sequence).	M
5	SDF_PNAME	Char(8)	Element parameter name i.e. name of the command parameter (CDF_PNAME) if the sequence element is a command or name of the sequence parameter (CSP_FPNAME) if the sequence element is a nested sequence.	M
6	SDF_FTYPE	Char(1)	Flag controlling whether the parameter value can be modified after loading the sequence on the stack or if it should be considered as fixed. Note that the element parameter could in principle be associated to a formal parameter (in which case its value can only be defined prior to the sequence loading as the formal parameter value and not modified any longer after loading the sequence). 'F' - if the parameter is fixed, i.e. once loaded on the stack the user cannot change its value. 'E' - if parameter is editable.	'E'

Fi. Nr	Field name	Field Type	Description	Ma/Def
7	SDF_VTYPE	Char(1)	<p>Flag identifying the source of the element parameter value and how to interpret the SDF_VALUE field.</p> <p>'R' - if the element parameter value has to be taken from SDF_VALUE and must be interpreted as raw.</p> <p>'E' - if the element parameter value has to be taken from SDF_VALUE and must be interpreted as engineering. This option can only be used for element parameters which are associated to a calibration curve (i.e. CPC_CATEG/ CSP_CATEG = 'T' or 'C').</p> <p>'F' - if the element parameter is associated to a sequence formal parameter whose name is specified in SDF_VALUE (CSP_FPNAME of this current sequence). This option cannot be used for command parameters of type 'group repeater' (CDF_GRPsize>0).</p> <p>'P' - if the element parameter value has to be taken from the corresponding entry in the parameter value set referenced by SDF_VALSET.</p> <p>'S' - if this is a sequence parameter and its value should be taken from the default value as specified for it in the CSP table (CSP_DEFVAL corresponding to the nested sequence's formal parameter). Note that if this option is used, the field CSP_DEFVAL cannot be left Null i.e. it must contain a value.</p> <p>'D' - if this is a command parameter and its value should be taken from the default value as specified in the corresponding CDF entry (CDF_VALUE). Note that if this option is used, it must be ensured that the corresponding CDF entry contains either directly (in CDF_VALUE) or indirectly (in CPC_DEFVAL) the value to be used for this parameter. This means that if option 'R' or 'E' is used for the field CDF_INTER in the corresponding CDF entry, the field CDF_VALUE cannot be left Null i.e. it must contain a value. If option 'D' is used for the field CDF_INTER in the corresponding CDF entry (default value to be extracted from the CPC_DEFVAL field), this latter field cannot be left Null i.e. it must contain a value. If option 'T' is used for the field CDF_INTER (default value to be extracted from a specified monitoring parameter), this implies that the default value for this command parameter will also be extracted from telemetry (after loading the sequence on a command source).</p>	M

Fi. Nr	Field name	Field Type	Description	Ma/Def
8	SDF_VALUE	Char(17)	<p>This field contains the value for the element parameter.</p> <p>If SDF_VTYPE is 'R' or 'E', this value must be expressed using the appropriate representation. Raw values must be expressed in a format compatible with the type of the parameter (as specified by CPC_PTC/CSP_PTC and CPC_RADIX/CSP_RADIX for unsigned integer). Engineering values must be expressed in a format compatible with the engineering representation of the parameter (CPC_DISPFORMAT/CSP_DISPFORMAT).</p> <p>If SDF_VTYPE is 'F', this field contains the name of the formal parameter (CSP_FPNAME) which this element parameter will inherit the value from.</p> <p>This field can be left Null only if SDF_VTYPE is set to 'D', 'P' or 'S'.</p>	
9	SDF_VALSET	Char(8)	<p>Name of the parameter value set identifier (PSV_PVSID) which will be used to provide the value.</p> <p>This field is only applicable if SDF_VTYPE='P', in which case it cannot be left Null.</p>	

3.3.3.4 Command sequence formal parameters: csp

This table describes the formal parameters associated to a command sequence. One record per formal parameter.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CSP_SQNAME	Char(8)	Name of the sequence which this is a formal parameter of (CSF_SQNAME).	M
2 *	CSP_FPNAME	Char(8)	Name of this formal parameter. The formal parameter name must be unique within each sequence but not necessarily unique across sequences.	M
3	CSP_FPNUM	Number(3)	Unique number of the formal parameter within the sequence. It permits reordering of parameters as required. Sequences cannot contain gaps in the numbering of their formal parameters, therefore consecutive numbers must be specified.	M
4	CSP_DESCR	Char(24)	Textual description of the formal parameter.	

Fi. Nr	Field name	Field Type	Description	Ma/Def
5	CSP_PTC	Number(2)	<p>Parameter Type Code. It identifies the parameter type i.e. unsigned integer, integer, real, octet string, character string, absolute time, relative time. Parameter Type Codes supported by SCOS-2000 and their interpretation are specified in Appendix A. Appendix A also specifies the format in which the raw values have to be specified for each supported parameter type.</p> <p>This field must be equivalent to the CPC_PTC/CSP/PTC of any associated element parameter (i.e. any element parameter referring to this formal parameter in the field SDF_VALUE).</p>	M
6	CSP_PFC	Number(5)	<p>Parameter Format Code. Along with the Parameter Type Code (CSP_PTC) this field identifies the parameter length (see Appendix A).</p> <p>Integer value in a range compatible with the specified PTC as specified in Appendix A.</p> <p>This field must be equivalent to the CPC_PFC/CSP_PFC of any associated element parameter.</p>	M
7	CSP_DISPFMT	Char(1)	<p>Flag controlling the input and display format of the engineering values for calibrated parameters and for time parameters.</p> <p>'A' – ASCII (if CSP_CATEG='T').</p> <p>'I' – signed integer (if CSP_CATEG='C' and CCA_ENGFMT='I')</p> <p>'U' – unsigned integer (if CSP_CATEG='C' and CCA_ENGFMT='U')</p> <p>'R' – real (if CSP_CATEG='C' and CCA_ENGFMT='R')</p> <p>'T' – absolute time (if CSP_PTC=9)</p> <p>'D' – delta/relative time (if CSP_PTC=10).</p> <p>The formats in which values are to be input/displayed for the different representation types are specified in Appendix A.</p> <p>This field must be equivalent to the CPC_DISPFMT/CSP_DISPFMT of any associated element parameter.</p>	'R'
8	CSP_RADIX	Char(1)	<p>This is the default radix that the raw representation of the parameter if it is unsigned integer.</p> <p>'D' – Decimal</p> <p>'H' – Hexadecimal</p> <p>'O' – Octal</p> <p>Note that the radix associated to this parameter is allowed to be different than the one (CPC_RADIX/CSP_RADIX) of the associated element parameter(s).</p>	'D'

Fi. Nr	Field name	Field Type	Description	Ma/Def
9	CSP_TYPE	Char(1)	<p>Flag identifying the type of the formal parameter.</p> <p>'C' - if this formal parameter is a command name (future extension)</p> <p>'S' - if this formal parameter is a sequence name (future extension)</p> <p>'P' - if this formal parameter is associated to one or more element parameters.</p>	M
10	CSP_VTYPE	Char(1)	<p>Flag identifying the representation used to express the formal parameter default value.</p> <p>'R' - if the default value (CSP_DEFVAL) is expressed in a raw representation.</p> <p>'E' - if the default value (CSP_DEFVAL) is expressed in an engineering representation. This option can only be used for parameters which are associated to a calibration curve (i.e. CSP_CATEG = 'T' or 'C').</p> <p>This field cannot be left Null for a formal parameter of type 'P' in case a default value is specified for it in CSP_DEFVAL. <i>It must be left Null if CSP_TYPE is 'C' or 'S'.</i></p>	
11	CSP_DEFVAL	Char(17)	<p>This field contains the default value for this formal parameter.</p> <p>For formal parameters of type 'P', this value must be expressed using the appropriate representation as specified in CSP_VTYPE. Raw values must be expressed in a format compatible with the type of the parameter (as specified by CSP_PTC and CSP_RADIX for unsigned integer). Engineering values must be expressed in a format compatible with the engineering representation of the parameter (CSP_DISPFORMAT).</p> <p><i>If CSP_TYPE is 'C' this must be a valid command ID (CCF_CNAME).</i></p> <p><i>If CSP_TYPE is 'S' this must be a valid sequence ID (CSF_SQNAME).</i></p> <p>This field maybe left Null indicating that the user MUST enter a value before loading the sequence.</p>	
12	CSP_CATEG	Char(1)	<p>Flag identifying the type of calibration associated to the parameter.</p> <p>'C' - numeric calibration, therefore CSP_CCAREF shall not be Null.</p> <p>'T' - textual calibration, therefore CSP_PAFREF shall not be Null.</p> <p>'B' - both calibrations (future extension).</p> <p>'N' - implies no calibration.</p> <p>This field must be left Null or set to 'N' for formal parameters whose raw representation is not of a numeric nature (i.e. strings, times).</p>	'N'

Fi. Nr	Field name	Field Type	Description	Ma/Def
13	CSP_PRFPREF	Number(4)	<p>This field contains the ID of an existing parameter range set (PRF_NUMBR) to which the parameter will be associated. The ranges defined within the set will determine the possible values for the parameter. The format of the selected range set (PRF_DSPFMT) must be compatible with the type of the command parameter (described by CSP_PTC or CSP_DISPFMT depending on the used representation i.e. PRF_INTER).</p> <p><i>This field must be Null if CSP_TYPE is 'C' or 'S'</i></p> <p>Note that the selected range set does not necessarily need to be the same one as specified for any associated element parameter. However, it should be considered that it is only safe to associate different range sets if they are more restrictive i.e. if the associated allowed ranges of values are all permitted for any associated element parameter.</p>	
14	CSP_CCAREF	Number(4)	<p>This field contains the ID of an existing numeric calibration curve set (CCA_NUMBR) to which the parameter will be associated. The value formats of the selected curve set (CCA_RAWFMT and CCA_ENGFMT) must be compatible with the type of the command parameter (described by CSP_PTC) and its engineering format (CSP_DISPFMT).</p> <p>It must be Null if CSP_CATEG is Null, 'N' or 'T'. It cannot be Null if CSP_CATEG is 'C' or 'B'. <i>It must be Null if CSP_TYPE is 'C' or 'S'.</i></p> <p>If not Null, this field must reference the same calibration curve as used by any associated element parameter (CPC_CCAREF/CSP_CCAREF).</p>	
15	CSP_PAFREF	Number(4)	<p>This field contains the ID of an existing textual calibration set (PAF_NUMBR) to which the parameter will be associated. The aliases contained within the set may be used as default values for the parameter. The selected alias set (PAF_RAWFMT) must be compatible with the type of the parameter (described by CSP_PTC).</p> <p>It must be Null if CSP_CATEG is Null, 'N' or 'C'. It cannot be Null if CSP_CATEG is 'T' or 'B'. <i>It must be Null if CSP_TYPE is 'C' or 'S'.</i></p> <p>If not Null, this field must reference the same textual calibration as used by any associated element parameter (CPC_PAFREF/CSP_PAFREF).</p>	
16	CSP_UNIT	Char(4)	<p>Engineering unit mnemonic e.g. 'VOLT' to be displayed besides formal parameter values. It should be consistent with the engineering unit (CPC_UNIT/CSP_UNIT) of any element parameter associated to this formal parameter.</p>	

3.3.3.4 Verification

SCOS-2000 supports two different types of verification checks: report based (i.e. based on dedicated TM packets reporting the successful or failed completion of a command execution stage) or parameter based (i.e. based on the check of the value of one or more monitoring parameters against a specified criteria). The characteristics of a verification stage are specified in the CVS table. Each command can be associated (in the CVP table) to several independent verification stages of different types. The verification stages based on parameter checks are associated to one or more verification criteria expressed in the CVE table.

Note that the same verification checks definition could in principle be associated to command sequences. However, the first SCOS-2000 release will only support verification checks for commands.

3.3.3.4.1 Verification stages file: cvs

This table defines the characteristics of a verification stage. One record per verification stage. The same verification stages can be associated to several commands.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CVS_ID	Number(5)	Unique ID of the verification stage. Integer value in the range (0....32767).	M
2	CVS_TYPE	Char(1)	The stage in the command execution profile to which the verification stage is associated. 'A' - application acceptance 'S' - start of execution '0' to '9' - progress number (step -n) of execution 'C' - completion of execution.	M
3	CVS_SOURCE	Char(1)	Flag identifying the type of verification check associated to this stage. 'R' - if based on TM report packets (see [RD-2]) 'V' - if based on the check of monitoring parameter values against the criteria specified in the CVE table for this stage.	M
4	CVS_START	Number(5)	Delta time after release (or execution for time tagged commands) before verification window opens. Positive integer number expressed in seconds.	M
5	CVS_INTERVAL	Number(5)	Length of the sampling interval used to calculate the overall verification window (as specified in [AD-2]). Positive integer number expressed in seconds.	M

6	CVS_SPID	Number(10)	<p>The SCOS-2000 Packet ID (PID_SPID) of the TM packet used to identify when to close a verification stage after the elapsing of its verification window (future extension).</p> <p>This field can only be specified if CVS_SOURCE is 'V'. It may be useful in the case of verification parameters which are extracted from several different TM packets. If not specified, the verification stage is closed as soon as any valid sample of any associated verification parameter is received with a time stamp during or after the verification window closure time (see [AD-2]).</p> <p>This field is irrelevant if CVS_SOURCE is 'R' (as the TM packets used for report based verification are implicitly known based on their type/subtype).</p>
---	----------	------------	--

3.3.3.4.2 Verification expressions: cve

This table describes the different monitoring parameters and values used to build the verification criteria for any defined verification stage. One record per verification check. Multiple checks associated to the same stage are implicitly ANDed together to form the overall verification criteria.

Remark: it is recommended that all the verification parameters associated to the same verification stage are extracted from (or calculated based upon) the same telemetry packet (see explanatory note of the CVS_SPID field above).

Remark: the validity of the monitoring parameters used for verification is implicitly taken into account by SCOS-2000 i.e. a verification check is only passed if the verification parameter is valid and its value satisfies the specified criterion.

This table is also used to identify the monitoring parameters which are affected by the execution of a command in order to disable their status consistency checking during the relevant verification window. Note that, in case the verification check is expressed in engineering form (CVE_INTER = 'E'), then also the OOL checks for the same parameter must be expressed in engineering form (OCF_INTER = 'C'). Further, TM parameters used for echo verification (CVE_INTER='C') should not be subject to status consistency checking (as one does not know which representation is going to be used to set the command parameter value and thus whether this will match the OCF definition for the affected TM parameter).

Fi. Nr	Field name	Field Type	Description	Ma/Def
1*	CVE_CVSID	Number(5)	ID of the verification stage (CVS_ID) this expression belongs to. It can only be a stage associated to parameter based verification checks (CVS_SOURCE = 'V').	M

2 *	CVE_PARNAM	Char(8)	Name of an existing monitoring parameter (PCF_NAME) to be used for the verification check. Any type of monitoring parameters can be used for verification i.e. TM parameters or synthetic parameters. However, only monitoring parameters whose raw representation is of numerical nature can be used (i.e. no strings, times). In case this TM parameter is subject to status consistency checking (PCF_USCON='Y'), then the representation used to specify the verification value (CVE_INTER) shall match with the representation used to specify the OOL checks (OCF_INTER).	M
3	CVE_INTER	Char(1)	Flag identifying the representation selected to specify the expected value (CVE_VAL) and the allowed tolerance (CVE_TOL) to be used for the verification check. 'R' - if the values (CVE_VAL & CVE_TOL) are expressed using a raw representation. Note that in case of synthetic parameters the raw representation is the source one i.e. the one resulting from the associated OL or hard-coded definition. 'E' - if the values (CVE_VAL & CVE_TOL) are expressed using an engineering representation (i.e. calibrated values). This option can only be used for verification parameters which are associated to a calibration curve (i.e. PCF_CURTX is not Null). 'C' - if the value (CVE_VAL) is a command parameter name. In this case the CVE_TOL will always be given using a raw representation. Thus, in case the verification parameter (CVE_PARNAM) is status consistency checked, it must be ensured that the raw representation is used to check it (OCF_INTER = 'U').	'R'
4	CVE_VAL	Char(17)	This field contains the value that the verification parameter (CVE_PARNAM) must be compared against in order to perform the verification check. This value must be expressed using the appropriate representation i.e. raw unless option 'E' is used in CVE_INTER. Raw values must be expressed in decimal in a format compatible with the type of the verification parameter (as specified by PCF_PTC). Engineering values must be expressed in a format compatible with the engineering representation of the verification parameter (i.e. ASCII if PCF_CATEG = 'S' or as specified in the associated CAF_ENGFMT for numerically calibrated parameters). In case of echo verification (CVE_INTER), this field contains the name of the command parameter (CDF_PNAME) whose value at command instantiation is used to check against to perform the verification check. Note that this parameter should only appear once in the command definition. If not, then the value of the first instance will be taken. This field is only applicable if CVE_CHECK='B' (i.e. if this entry is actually used for verification purposes), in which case it cannot be left Null. This field is not used if CVE_CHECK='S'.	

5	CVE_TOL	Char(17)	<p>This field contains the allowed tolerance of the verification parameter value to satisfy the verification check (i.e. the value of CVE_PARNAM must fall within the range identified by CVE_VAL +/- CVE_TOL). The tolerance must be specified in a format compatible with the required representation (see CVE_INTER) of the verification parameter in the same way as for CVE_VAL</p> <p>This field must be left Null in case an engineering representation (CVE_INTER = 'E') is used for a status verification parameter (PCF_CATEG = 'S').</p>	
6	CVE_CHECK	Char(1)	<p>Flag controlling whether this check has to be used only to drive the status consistency checking of the verification parameter or also for the actual command verification.</p> <p>'B' – if this check has to be used both for actual verification and to disable/re-enable status consistency checking of CVE_PARNAM (if this happens to be a status consistency checked parameter). Using this option, the value specified in CVE_VAL is used to establish the new expected value at the end of the verification stage window.</p> <p>'S' – if this check has to be used only to disable status consistency checking of CVE_PARNAM. Note that this option can only be used in conjunction with one or more other verification checks associated to the same verification stage with CVE_CHECK set to 'B'.</p>	'B'

3.3.3.4.3 Verification profiles: cvp

This table describes which stages are applicable to which command/sequence, i.e. the verification profile of each task. One record per verification stage attached to a command/sequence. The first release of SCOS-2000 will only support verification for commands. Several verification stages can be associated to each command. However, only one verification stage for each CVS_TYPE (e.g. only one acceptance verification stage can be associated to a command).

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CVP_TASK	Char(8)	This field contains the name of the command or of the sequence (CCF_CNAME or CSF_NAME) using the verification stage specified in CVP_CVSID.	M
2 *	CVP_TYPE	Char(1)	<p>Flag identifying the type of the task associated to the verification stage.</p> <p>'C' - if the task is a command,</p> <p>'S' - if the task is a sequence (not supported by SCOS-2000).</p>	'C'

Fi. Nr	Field name	Field Type	Description	Ma/Def
3 *	CVP_CVSID	Number(5)	This field contains the ID (CVS_ID) of the stage associated to the task. Note that there must be at most one verification stage of a specified type (e.g. CVS_TYPE = 'A' or 'C') associated to each command.	M

3.3.3.5 Command/Sequence Parameter Sets

The following tables describe the definition of parameter sets. These are collection of command or sequence parameters for which predefined combination of values (parameter value sets) can be specified in the database. The parameter value sets can then be applied in the real-time environment to specify a value for all contained parameters using high-level mnemonics. Parameter sets and parameter value sets can be specified for commands and sequences.

3.3.3.5.1 Parameter sets: pst

This table defines the parameter sets i.e. named collection of command or sequence parameters. One record per parameter set.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PST_NAME	Char(8)	Unique name of the parameter set.	M
2	PST_DESCR	Char(24)	Textual description of the parameter set.	

3.3.3.5.2 Parameter value sets: psv

This table defines the parameter value sets i.e. the name and description of the named set of values for the parameters belonging to a set. One record per parameter value set.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1	PSV_NAME	Char(8)	Name of the parameter set (PST_NAME) to which this parameter value set is associated.	M
2 *	PSV_PVSID	Char(8)	Unique name of the parameter value set.	M
3	PSV_DESCR	Char(24)	Textual description of the parameter value set.	

3.3.3.5.3 Parameter sets definition: cps

This table can be used to explicitly specify the parameters contained in a parameter set. One record per parameter belonging to a parameter set. It is not used by SCOS-2000 (parameter sets are defined in the pst table, see Section 3.3.3.5.1 above) but it can be useful in the off-line database system to check the completeness of the associated parameter value sets.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CPS_NAME	Char(8)	Name of a parameter set (PST_NAME).	
2	CPS_PAR	Char(8)	Name of a command parameter (CDF_PNAME) or a formal parameter name (CSP_FPNAME) belonging to the parameter set specified in CPS_NAME.	
3 *	CPS_BIT	Number(4)	For command parameters, this field shall contain its bit offset (CDF_BIT) for unique identification in the set. For sequence parameters, this field shall contain its position (CSP_FPNUM).	

3.3.3.5.4 Parameter value sets definition: pvs

This table defines the values contained in parameter value sets i.e. the value to be given to each contained parameter when applying the parameter value set. One record per value belonging to a parameter value set.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PVS_ID	Char(8)	Name of the parameter value set (PSV_PVSID) to which this record applies.	M
2	PVS_PSID	Char(8)	Name of the parameter set to which the parameter value set belongs to (PST_NAME).	M
3	PVS_PNAME	Char(8)	Name of the command parameter (CDF_PNAME) or sequence parameter (CSP_FPNAME) for which a value is specified in PVS_VALS. Note: if two sequence parameters or a command and a sequence parameter are given the same name and they are used within parameter sets, it must be ensured that they are associated to the same raw and engineering formats i.e. to the same CPC_PTC/CSP_PTC and to the same CPC_DISPFMT/CSP_DISPFMT.	M
4	PVS_INTER	Char(1)	Flag identifying the representation selected to express the value (PVS_VALS) for this parameter. 'R' - if the value is expressed using a raw representation. 'E' - if the value is expressed using an engineering representation. This option can only be used for parameters which are associated to a calibration curve (i.e. CPC_CATEG/CSP_CATEG = 'T' or 'C').	'R'

Fi. Nr	Field name	Field Type	Description	Ma/Def
5	PVS_VALS	Char(17)	The value corresponding to the parameter identified by PVS_PNAME. The parameter value must be expressed using the appropriate representation i.e. raw unless option 'E' is used in PVS_INTER. Raw values must be expressed in a format compatible with the type of the parameter (as specified by CPC_PTC/CSP_PTC and CPC_RADIX/CSP_RADIX for unsigned integer). Engineering values must be expressed in a format compatible with the engineering representation of the parameter (CPC_DISPFORMAT/CSP_DISPFORMAT). This field can be left Null in which case the current value for this parameter is unaffected when applying this parameter value set.	
6 *	PVS_BIT	Number(4)	For command parameters, this field shall contain its bit offset (CDF_BIT) for unique identification in the set. For sequence parameters, this field shall contain its position (CSP_FPNUM).	M

3.3.3.5.5 Parameter sets mapping to commands/sequences: psm

This table defines which parameter sets are used by which commands or sequences. The same parameter set can be associated to several commands or sequences. However, it is not possible to associate the same parameter set to a command and a sequence (this because even if a command and a sequence may have the same set of parameters with the same name, the field PVS_BIT would unavoidably have different values when creating parameter value sets).

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PSM_NAME	Char(8)	The name of the task i.e. command (CCF_CNAME) or sequence (CSF_NAME) to which the parameter set specified in PSM_PARSET is associated.	M
2 *	PSM_TYPE	Char(1)	Flag identifying the type of the task associated to the parameter set. 'C' - if the task is a command, 'S' - if the task is a sequence.	M
3 *	PSM_PARSET	Char(8)	Name of the parameter set (PST_NAME) associated to this task.	M

3.3.3.6 Command/Sequence Parameters (De-)calibrations

The following tables define the data required to perform conversions of the parameter values. Unlike the tables used to calibrate monitoring parameters, the commanding calibrations can be used in both directions i.e. to convert values from engineering to raw (for display and encoding purposes) or from raw to engineering (for display purposes). They can be equally associated to command as well as to sequence parameters. However, only parameters which in their raw representation are of a numerical nature can be associated to a calibration (i.e. no strings, times).

3.3.3.6.1 Numerical (de-)calibration curves: cca

This table defines the numerical (de-)calibration curve sets. One record per (de-)calibration curve.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CCA_NUMBR	Number(4)	Numerical (de-)calibration curve identification number. Integer value in the range (1...9999) uniquely identifying the commanding numerical (de-)calibration curve.	M
2	CCA_DESCR	Char(24)	Textual description of the parameter curve set.	
3	CCA_ENGFMT	Char(1)	Flag identifying the format type for the engineering values specified in the corresponding records of the CCS table. ‘I’ – signed integer ‘U’- unsigned integer ‘R’- real This format has to be compatible with the engineering format of the parameters associated to this (de-)calibration curve.	‘R’
4	CCA_RAWFMT	Char(1)	Flag identifying the format type for the raw values specified in the corresponding records of the CCS table. ‘I’ – signed integer ‘U’- unsigned integer ‘R’- real This format must be compatible with the type of the parameters associated to this (de-)calibration curve.	‘U’
5	CCA_RADIX	Char(1)	Flag identifying the radix used for the raw values (CCS_YVALS) specified in the corresponding records of the CCS table. Only applicable for unsigned integer values (i.e. CCA_RAWFMT = ‘U’). ‘D’ – Decimal ‘H’ – Hexadecimal ‘O’ – Octal Note that the radix associated to this calibration curve is allowed to be different than the one (CPC_RADIX/CSP_RADIX) of the command/sequence parameter(s) using it.	‘D’
6	CCA_UNIT	Char(4)	Engineering unit mnemonic. Available for consistency checking against the unit of any associated command or sequence parameter (CPC_UNIT/CSP_UNIT).	
7	CCA_NCURVE	Number(3)	Number of records defined in the CCS table for this curve.	

3.3.3.6.2 Numerical (de-)calibration curves definition: ccs

This table defines the (de-)calibration curve value pairs. One record per engineering/raw value pair.

Remark: note that the engineering values in this table are specified in the CCA_XVALS field (unlike the calibration curve tables used for monitoring parameters calibrations).

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	CCS_NUMBR	Number(4)	Identification number (CCA_NUMBR) of the (de-)calibration curve to which this value pair belongs.	M
2	CCS_XVALS	Char(17)	Engineering value expressed in a format compatible with the corresponding CCA definition (CCA_ENGFMT). All values including unsigned integer values have to be expressed in decimal regardless of the radix associated to this calibration curve (CCA_RADIX).	M
3 *	CCS_YVALS	Char(17)	Raw value expressed in a format compatible with the corresponding CCA definition (CCA_RAWFMT). Unsigned integer values (CCA_RAWFMT = 'U') have to be expressed using the radix associated to this calibration curve (CCA_RADIX).	M

3.3.3.6.3 Textual (de-)calibrations: paf

This file defines the text (de-)calibration sets (also known as parameter alias sets). One record per textual (de-)calibration set.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PAF_NUMBR	Number(4)	Textual (de-)calibration set identification number. Integer value in the range (1...9999) uniquely identifying the commanding textual (de-)calibration set.	M
2	PAF_DESCR	Char(24)	Textual description of the (de-)calibration set.	
3	PAF_RAWFMT	Char(1)	Flag identifying the format type for the raw values specified in the corresponding records of the PAS table below. 'I' – signed integer 'U' - unsigned integer 'R' - real This format has to be compatible with the type of the parameters associated to this textual (de-)calibration.	'U'
4	PAF_NALIAS	Number(3)	Number of records defined in the PAS table for this set.	

3.3.3.6.4 Textual (de-)calibrations definitions: pas

This table defines the text string/raw value pairs associated to a (de-)calibration set. One record per alias/raw value pair.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PAS_NUMBR	Number(4)	Identification number (PAF_NUMBR) of the textual (de-)calibration set to which this string/value pair belongs.	M
2	PAS_ALTXT	Char(16)	This field contains the text string used to provide an engineering interpretation to the raw value specified in PAS_ALVAL.	M
3 *	PAS_ALVAL	Char(17)	Raw value expressed (in decimal) in a format compatible with the corresponding PAF definition (PAF_RAWFMT).	M

3.3.3.7 Command/Sequence Parameters Range Checks

The following tables define the data required to perform range checking of parameter values. They can be equally used for command as well as for sequence parameters. However, only parameters which in their raw representation are of a numerical nature or times can be range checked (i.e. strings cannot be range checked).

3.3.3.7.1 Parameter range sets: prf

This table describes the set of ranges which can be associated to a command or a sequence parameter. One record per range set.

Remark: when loading command sequences on a manual stack, only the formal parameter values are checked against the applicable range sets i.e. not all the values of any contained sequence/command. However, it is possible to configure SCOS-2000 such that all parameter values are checked at sequence loading time.

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PRF_NUMBR	Number(4)	Unique identification number of the parameter range set. Integer value in the range (1...9999).	M
2	PRF_DESCR	Char(24)	Textual description of the parameter range set.	
3	PRF_INTER	Char(1)	Flag identifying the representation used to express the range values for this range set in the PRV table. 'R' – if the range values (PRV_MINVAL and PRV_MAXVAL) are expressed using a raw representation. 'E' – if the range values (PRV_MINVAL and PRV_MAXVAL) are expressed using an engineering representation. This option can only be used if all the parameters referencing this range set are associated to a calibration curve (CPC_CATEG/CSP_CATEG='T' or 'C').	'R'

4	PRF_DSPFMT	Char(1)	<p>Flag identifying the representation type of the values specified for this range set in the PRV table.</p> <p>'A' – ASCII 'I' – signed integer 'U' – unsigned integer 'R' – real 'T' – absolute time 'D' – delta/relative time.</p> <p>The formats in which values for the different representation types have to be expressed are specified in Appendix A.</p> <p>Note that this representation type must be compatible with the one of any associated parameter (described in CPC_PTC/CSP_PTC if PRF_INTER='R' and in CPC_DISPFORMAT/CSP_DISPFORMAT if PRF_INTER='E' or for time parameters).</p>	'U'
5	PRF_RADIX	Char(1)	<p>Flag identifying the radix used for the range values specified in the corresponding records of the PRV table. Only applicable for unsigned integer values (i.e. PRF_DSPFMT='U').</p> <p>'D' – Decimal 'H' – Hexadecimal 'O' – Octal</p> <p>Note that the radix associated to this range set is allowed to be different than the one (CPC_RADIX/CSP_RADIX) of the command/sequence parameter(s) using it.</p>	'D'
6	PRF_NRANGE	Number(3)	Number of records defined in the PRV table for this range set.	
7	PRF_UNIT	Char(4)	Engineering unit mnemonic. Available for consistency checking against the unit of any associated command or sequence parameter (CPC_UNIT/CSP_UNIT).	

3.3.3.7.2 Parameter range values: prv

This table defines the allowed value ranges for the associated range set. One record per allowed (range of) value(s). The range check will be satisfied if the parameter values falls within or is equal to the min/max values. A set of allowed discrete values can be defined by entering a record per allowed value (specified in PRV_MINVAL).

Fi. Nr	Field name	Field Type	Description	Ma/Def
1 *	PRV_NUMBR	Number(4)	Identification number (PRF_NUMBR) of the range set to which this allowed value range belongs.	M

Fi. Nr	Field name	Field Type	Description	Ma/ Def
2 *	PRV_MINVAL	Char(17)	This field contains the low limit value (or the allowed value if PRV_MAXVAL is Null). This value must be expressed in a format compatible with the corresponding PRF definition (PRF_DSPFMT and also PRF_RADIX for unsigned integers).	M
3	PRV_MAXVAL	Char(17)	This field contains the high limit value. It can be left Null indicating that the low value is a discrete value. This value must be expressed in a format compatible with the corresponding PRF definition (PRF_DSPFMT and also PRF_RADIX for unsigned integers).	

Appendix A – SCOS-2000 Parameter Types

The table below lists the Parameter Type and Format Codes supported by SCOS-2000. It also specifies the parameter type (internal format) and the parameter width assumed by the importer for each PTC. Reference is made to the nomenclature adopted in the Packet Utilisation Standard (see [RD-6]). PTC/PFC combinations supported by SCOS-2000 but not listed in [RD-6] are emphasized in bold. Rows in italics refer to PTC/PFC combinations listed in [RD-6] but not supported by SCOS -2000.

PTC	PFC	S2K Parameter Type	Length	PUS type + Comment
1	0	Unsigned Integer	1 bit	Boolean parameter
2	>0, <33	Unsigned Integer	PFC	Enumerated parameter. In the PUS only some PFCs are allowed for this parameter type. In SCOS-2000, the only restriction is the maximum parameter length (32 bits).
3	0	Unsigned Integer	4 bits	Unsigned integer parameter
3	1	Unsigned Integer	5 bits	Unsigned integer parameter
3	2	Unsigned Integer	6 bits	Unsigned integer parameter
3	3	Unsigned Integer	7 bits	Unsigned integer parameter
3	4	Unsigned Integer	8 bits	Unsigned integer parameter
3	5	Unsigned Integer	9 bits	Unsigned integer parameter
3	6	Unsigned Integer	10 bits	Unsigned integer parameter
3	7	Unsigned Integer	11 bits	Unsigned integer parameter
3	8	Unsigned Integer	12 bits	Unsigned integer parameter
3	9	Unsigned Integer	13 bits	Unsigned integer parameter
3	10	Unsigned Integer	14 bits	Unsigned integer parameter
3	11	Unsigned Integer	15 bits	Unsigned integer parameter
3	12	Unsigned Integer	16 bits	Unsigned integer parameter
3	13	Unsigned Integer	24 bits	Unsigned integer parameter
3	14	Unsigned Integer	32 bits	Unsigned integer parameter
3	<i>15</i>	<i>Unsigned Integer</i>	<i>48 bits</i>	<i>Unsigned integer parameter (not supported by SCOS-2000).</i>
3	<i>16</i>	<i>Unsigned Integer</i>	<i>64 bits</i>	<i>Unsigned integer parameter (not supported by SCOS-2000).</i>
4	0	Signed Integer	4 bits	Signed integer parameter
4	1	Signed Integer	5 bits	Signed integer parameter

PTC	PFC	S2K Parameter Type	Length	PUS type + Comment
4	2	Signed Integer	6 bits	Signed integer parameter
4	3	Signed Integer	7 bits	Signed integer parameter
4	4	Signed Integer	8 bits	Signed integer parameter
4	5	Signed Integer	9 bits	Signed integer parameter
4	6	Signed Integer	10 bits	Signed integer parameter
4	7	Signed Integer	11 bits	Signed integer parameter
4	8	Signed Integer	12 bits	Signed integer parameter
4	9	Signed Integer	13 bits	Signed integer parameter
4	10	Signed Integer	14 bits	Signed integer parameter
4	11	Signed Integer	15 bits	Signed integer parameter
4	12	Signed Integer	16 bits	Signed integer parameter
4	13	Signed Integer	24 bits	Signed integer parameter
4	14	Signed Integer	32 bits	Signed integer parameter
4	15	<i>Signed Integer</i>	<i>48 bits</i>	<i>Signed integer parameter (not supported by SCOS-2000).</i>
4	16	<i>Signed Integer</i>	<i>64 bits</i>	<i>Signed integer parameter (not supported by SCOS-2000).</i>
5	1	Simple precision real (IEEE standard)	32 bits	Simple precision real parameter
5	2	Double precision real (IEEE standard)	64 bits	Double precision real parameter
5	3	Simple precision real (MIL standard)	32 bits	32 bit floating point numbers encoded as per MIL STD 1750A (normally referred to as PTC=5, PFC=2 in ESA missions Parameter Types definitions).
5	4	Extended precision real (MIL standard)	48 bits	48-bit floating point numbers encoded as per MIL STD 1750A.
6	0	<i>Bit string</i>	<i>Variable</i>	<i>Variable-length bit-string (not supported by SCOS-2000).</i>
6	>0, <33	Unsigned integer	PFC bits	PUS bit-string parameters are handled by SCOS-2000 as unsigned integer parameters with the length given by their PFC (up to 32 bits). They are only handled on the monitoring side (i.e. not on the commanding side).

PTC	PFC	S2K Parameter Type	Length	PUS type + Comment
7	0	Octet string	Variable	Variable-length octet-string. This is only supported for command/sequence parameters (i.e. not for monitoring parameters). Values of variable octet string parameters can only be loaded from a Stack Import file (see [RD-12]) (i.e. they cannot be specified/edited on SCOS-2000 manual stacks). Nevertheless, octet string parameters have to be specified as Editable elements of commands (field CDF_ELTYPE= 'E') if one wants to be able to specify and import their value from stack files.
7	> 0	Octet string	PFC octets	Fixed-length octet-strings. These types are supported on both the monitoring and commanding side. Values of octet string parameters can only be specified in the database or loaded from a Stack Import file (see [RD-12]) (i.e. they cannot be specified/edited on SCOS-2000 manual stacks). Nevertheless, octet string parameters have to be specified as Editable elements of commands (field CDF_ELTYPE= 'E') if one wants to be able to specify and import their value from stack files.
8	0	ASCII string	Variable	Variable-length character-string. This is only supported for command/sequence parameters (i.e. not for monitoring parameters). Values of variable character string command/sequence parameters can only be loaded from a Stack Import file (see [RD-12]) (i.e. they cannot be specified/edited on SCOS-2000 manual stacks). Nevertheless, character string parameters have to be specified as Editable elements of commands (field CDF_ELTYPE= 'E') if one wants to be able to specify and import their value from stack files.
8	> 0	ASCII string	PFC octets	Fixed-length character-strings. These types are supported on both the monitoring and commanding side. Values for character string command/sequence parameters can only be specified in the database or loaded from an external file (see [RD-12] and [RD-13]) (i.e. they cannot be specified/edited in SCOS-2000 manual stacks). Nevertheless, character string parameters have to be specified as Editable elements of commands (field CDF_ELTYPE= 'E') if one wants to specify their value as part of external files.

PTC	PFC	S2K Parameter Type	Length	PUS type + Comment
9	0	Absolute time (CUC or CDS format) with p-field	Variable	Full definition of an absolute time based on its p-field (not supported by SCOS-2000). For the definition of these time formats see [RD-8].
9	1	Absolute time CDS format without microseconds (6 bytes without p-field)	6 octets	Absolute time CDS format without microseconds (not supported by SCOS-2000). For the definition of these time formats see [RD-8].
9	2	Absolute time CDS format with microseconds (8 bytes without p-field)	8 octets	Absolute time CDS format with micro seconds. CDS format is only supported for command parameters (not for monitoring parameters). For the definition of these time formats see [RD-8].
9	3	Absolute time CUC format (1 byte coarse time)	1 octet	Absolute time CUC format.
9	4	Absolute time CUC format (1 byte coarse time, 1 byte fine time)	2 octets	Absolute time CUC format
9	5	Absolute time CUC format (1 byte coarse time, 2 bytes fine time)	3 octets	Absolute time CUC format.
9	6	Absolute time CUC format (1 byte coarse time, 3 bytes fine time)	4 octets	Absolute time CUC format.
9	7	Absolute time CUC format (2 bytes coarse time)	2 octets	Absolute time CUC format.
9	8	Absolute time CUC format (2 bytes coarse time, 1 byte fine time)	3 octets	Absolute time CUC format.
9	9	Absolute time CUC format (2 bytes coarse time, 2 bytes fine time)	4 octets	Absolute time CUC format.
9	10	Absolute time CUC format (2 bytes coarse time, 3 bytes fine time)	5 octets	Absolute time CUC format.
9	11	Absolute time CUC format (3 bytes coarse time)	3 octets	Absolute time CUC format.
9	12	Absolute time CUC format (3 bytes coarse time, 1 byte fine time)	4 octets	Absolute time CUC format.
9	13	Absolute time CUC format (3 bytes coarse time, 2 bytes fine time)	5 octets	Absolute time CUC format.

PTC	PFC	S2K Parameter Type	Length	PUS type + Comment
9	14	Absolute time CUC format (3 bytes coarse time, 3 bytes fine time)	6 octets	Absolute time CUC format.
9	15	Absolute time CUC format (4 bytes coarse time)	4 octets	Absolute time CUC format.
9	16	Absolute time CUC format (4 bytes coarse time, 1 byte fine time)	5 octets	Absolute time CUC format.
9	17	Absolute time CUC format (4 bytes coarse time, 2 bytes fine time)	6 octets	Absolute time CUC format.
9	18	Absolute time CUC format (4 bytes coarse time, 3 bytes fine time)	7 octets	Absolute time CUC format.
10	3	Relative time CUC format (1 byte coarse time)	1 octet	Relative time CUC format. For the definition of these time formats see [RD-8].
10	4	Relative time CUC format (1 byte coarse time, 1 byte fine time)	2 octets	Relative time CUC format
10	5	Relative time CUC format (1 byte coarse time, 2 bytes fine time)	3 octets	Relative time CUC format.
10	6	Relative time CUC format (1 byte coarse time, 3 bytes fine time)	4 octets	Relative time CUC format.
10	7	Relative time CUC format (2 bytes coarse time)	2 octets	Relative time CUC format.
10	8	Relative time CUC format (2 bytes coarse time, 1 byte fine time)	3 octets	Relative time CUC format.
10	9	Relative time CUC format (2 bytes coarse time, 2 bytes fine time)	4 octets	Relative time CUC format.
10	10	Relative time CUC format (2 bytes coarse time, 3 bytes fine time)	5 octets	Relative time CUC format.
10	11	Relative time CUC format (3 bytes coarse time)	3 octets	Relative time CUC format.
10	12	Relative time CUC format (3 bytes coarse time, 1 byte fine time)	4 octets	Relative time CUC format.

PTC	PFC	S2K Parameter Type	Length	PUS type + Comment
10	13	Relative time CUC format (3 bytes coarse time, 2 bytes fine time)	5 octets	Relative time CUC format.
10	14	Relative time CUC format (3 bytes coarse time, 3 bytes fine time)	6 octets	Relative time CUC format.
10	15	Relative time CUC format (4 bytes coarse time)	4 octets	Relative time CUC format.
10	16	Relative time CUC format (4 bytes coarse time, 1 byte fine time)	5 octets	Relative time CUC format.
10	17	Relative time CUC format (4 bytes coarse time, 2 bytes fine time)	6 octets	Relative time CUC format.
10	18	Relative time CUC format (4 bytes coarse time, 3 bytes fine time)	7 octets	Relative time CUC format.
11	0	N/A	N/A	Deduced parameter. The parameter characteristics are derived for a specific instance of the parameter field from the value of another preceding parameter field in the same packet (see Section 3.3.2.5.2 above).
13	0	N/A	N/A	Saved synthetic parameter. The parameter value characteristics are derived from the related synthetic parameter (PCF_RELATED) (see Section 3.3.2.1.1 above).

Table 4 - Parameter Type and Format Codes supported by SCOS -2000

The table below lists the applicable encoding and raw formats for each parameter type supported by SCOS-2000. The encoding format is the one used automatically by the system to encode/decode the parameter values and obtain the raw value view. The raw formats are used by the system to e.g. syntactically check command/sequence parameter values. For parameters which in their raw representation are of a numerical nature, the engineering format is dictated by the associated calibration curve (field CAF_ENGFMT for monitoring parameters and CCA_ENGFMT for command/sequence parameters).

Type	Encoding format	Raw format	Examples	Comment
Unsigned integer	Unsigned integer	Positive integers only. Allowed digits depend on the select radix.	15, 1111, 17, F	Unsigned integer raw values can be specified/displayed in decimal (default), binary, octal or hexadecimal. Engineering (i.e. calibrated values) are always input/displayed in decimal even if unsigned integer (i.e. CAF_ENGFMT/CCA_ENGFMT='U').
Signed integer	2's complement (first bit used for the sign)	Positive and negative integers.	127, -1, -127	Only decimal is allowed.
Simple precision real (IEEE)	IEEE Std 754-1985 (see [RD-6])	IEEE scientific or fixed point notation.	23.13,-1.2E3, 2.5E38	
Double precision real (IEEE)	IEEE Std 754-1985 (see [RD-6])	IEEE scientific or fixed point notation.	23.13,-1.2E3, 1.6E308	
Single precision real (MIL)	MIL-STD-1750-A (see [RD-7])	IEEE scientific or fixed point notation.	23.13,-1.2E3, 1.6E38	This is referred to as PTC=5, PFC=2 in [RD-7].
Octet-string	Unsigned integer	ASCII digits in Hex (two digits per octet with no spacing)	0AAFBB	In the case of Variable Octet String parameters, the actual parameter length is determined by SCOS-2000 command sources by counting the number of octets in the value (1 octet every two digits) and not by reading the value of 'n' (number of octets) (see Section 23.5.7 of PUS [RD-6]). If required by the actual encoding of Variable Octet String parameters, the value of 'n' has to be explicitly included in the value itself (i.e. it is not added by SCOS-2000 command sources). Parameters of this type cannot be associated to a calibration curve.

Type	Encoding format	Raw format	Examples	Comment
Characterstring	ASCII string	ASCII characters	ANY-STRING	In the case of Variable Character String parameters, the actual parameter length is determined by SCOS-2000 command sources by counting the number of characters in the value (1 byte every character) and not by reading the value of 'n' (number of characters) (see Section 23.5.8 of PUS [RD-6]). If required by the actual encoding of Variable Character String parameters, the value of 'n' has to be explicitly included as a separate parameter or a mission specific modification is required (i.e. it is not added by SCOS-2000 command sources). Parameters of this type cannot be associated to a calibration curve.
Absolute time	See [RD-6].	yyyy.ddd.hh.mm.ss	2001.159.23.59.59	The conversion between the raw format and encoding format takes automatically place based on a mission specific time correlation routine. Parameters of this type cannot be associated to a calibration curve.
Relative time	See [RD-6].	+/-[ddd.]hh.mm.ss[mmm]	-00.01.00, +251.23.59.59.010	Parameters of this type cannot be associated to a calibration curve.

Table 5 - Value formats for the SCOS-2000 parameter types

Appendix B – SCOS-2000 Database Import Tables Overview

This Appendix provides an overview of all the tables imported into the SCOS-2000 run-time database along with the relationships between them. Arrows are used to depict multiple relationships.

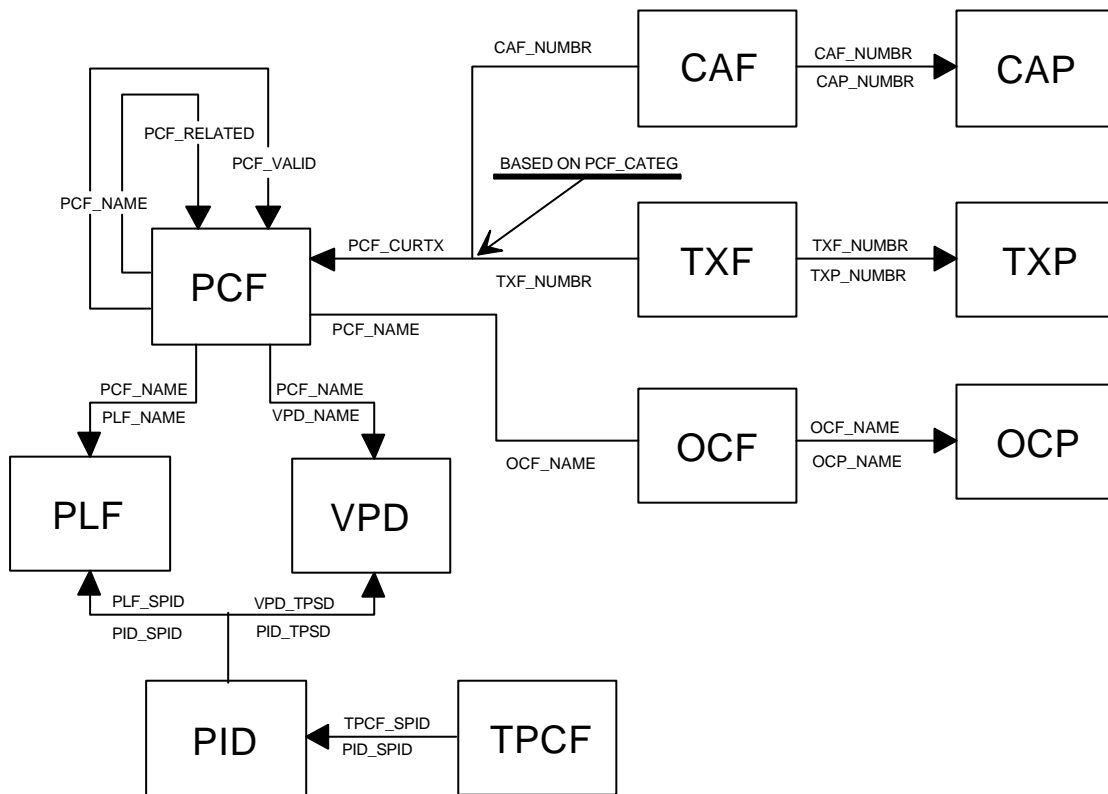
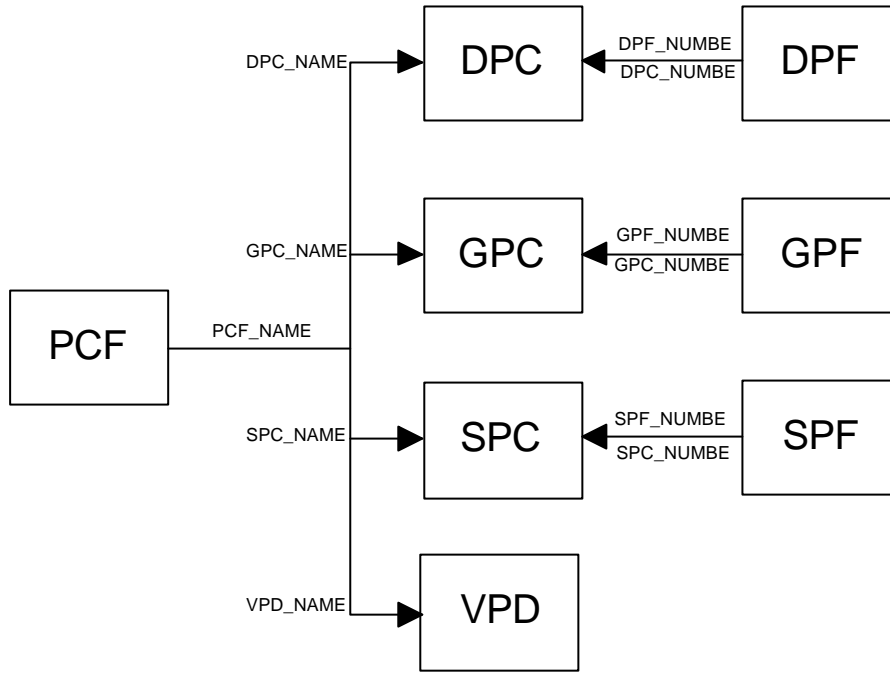
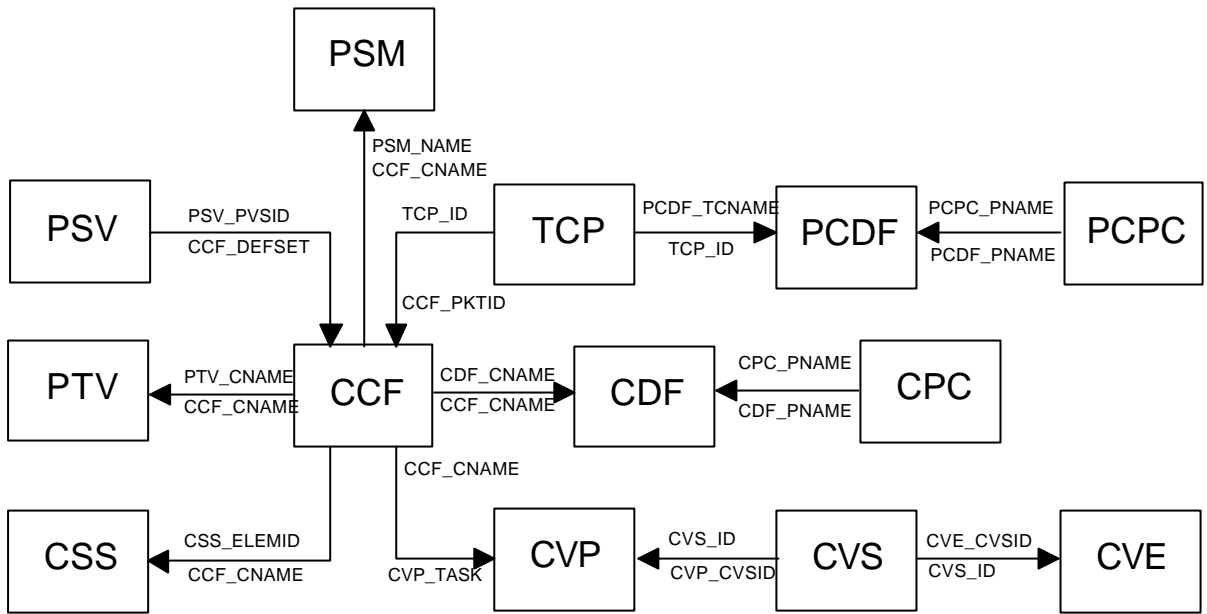


Figure 1 - Relationships for the Monitoring tables.



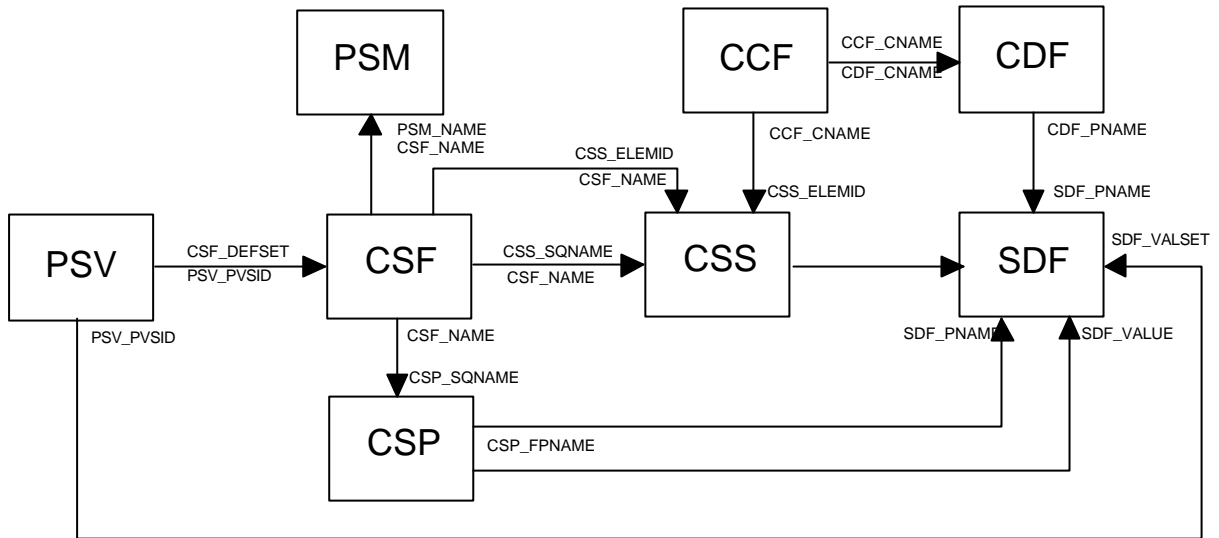
Relationships for the Display tables.ds

Figure 2 - Relationships for the Display tables



Relationships for the Command tables.dsi

Figure 3 - Relationships for the Command tables



Relationships for the Command Sequence tables.dsF

Figure 4 - Relationships for the Command Sequences tables

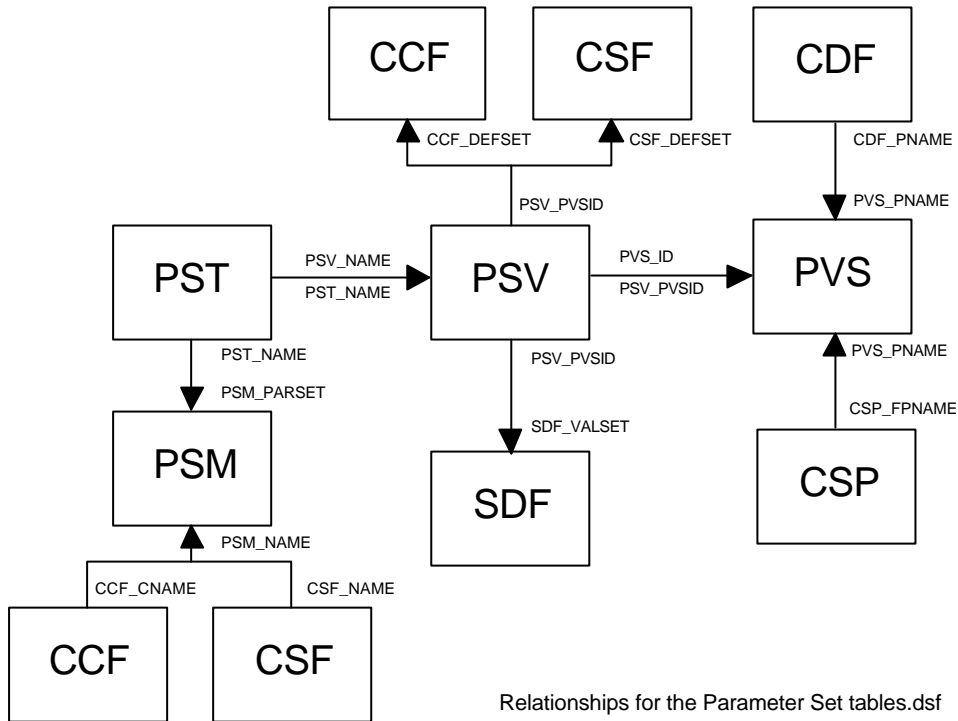
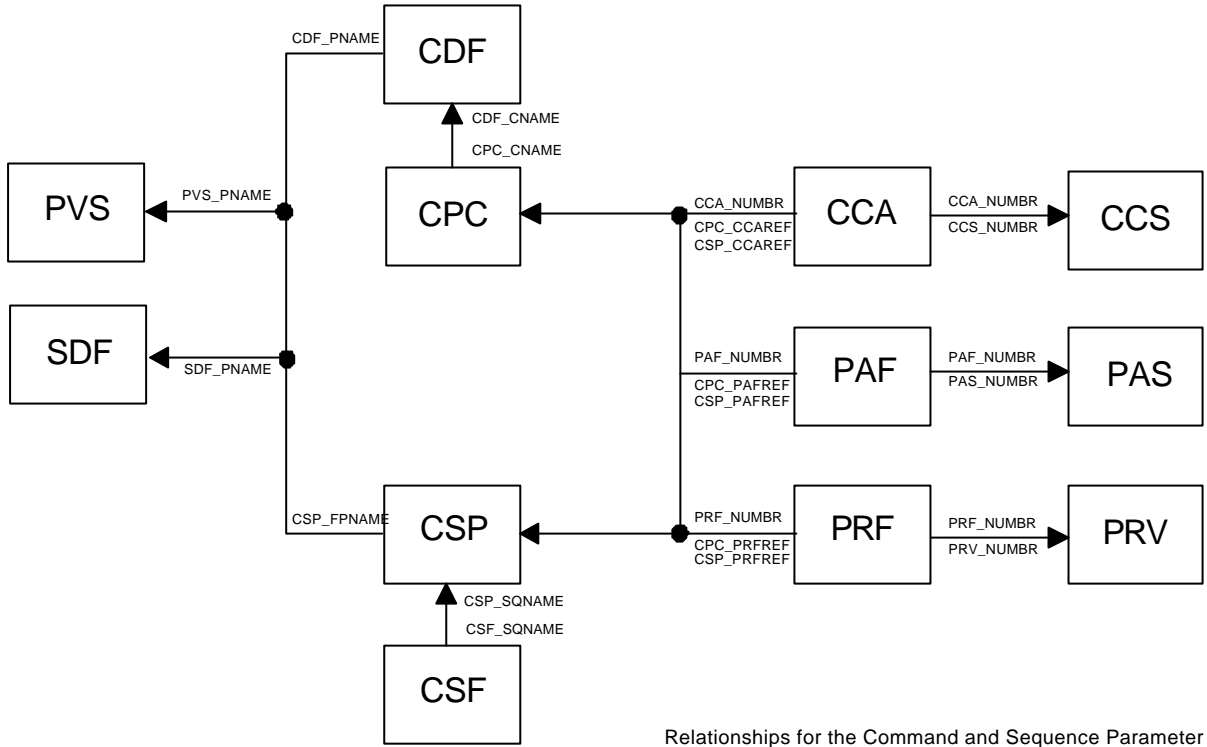


Figure 5 - Relationships for the Parameter Set tables



Relationships for the Command and Sequence Parameter 1

Figure 6 - Relationships for the Command and Sequence Parameter tables

Annex 1 – List of implemented Document Changes

General

DCR-237: add the parameter type (5,4) in order to support 48-bit real parameters encoded according to the Military Standard.

DCR-198: clarify if all values fitting within the maximum field length are allowed for fields e.g. CVS_SPID, PCD_PID, etc.

Monitoring

DCR-251: Align description of UDCs and applicable constraints to Database Import ICD V4.3 (consolidation version). This removes the ability to associate a single UDC to both the static and the dynamic views.

DCR-253: State that the allocation between packets identification fields (e.g. APID/Type/Subtype) and S2K SPIDs/TPSDs should not change. Further, PID entries containing SPIDs which have been used during a mission phase shall not be deleted. This is required in order to ensure appropriate working of the data retrieval functions (such as GDDS).

DCR-260: State that there should only be one valid combination of SPID/TPSD/APID/Type/Subtype in the PID Table. This is required in order to support proper filtering of the Packets History Display.

DCR-261: Explicitly introduce a constraint in the name of monitoring parameters (they are not allowed to start with the character "\$"). This is because the OL syntax uses the character "S" to identify static parameters.

Clarify the actual usage of the flags DCP_MODE and SPC_MODE (setting them to 'Y' enables the dynamic configuration option within the Telemetry Desktop allowing the enabling/disabling of invalid values display).

DCR-262: add the field VPD_OFFSET to the VPD table in order to enable the definition of VDF entries containing gaps and overlaps.

Add remark stating that only the parameters contained within packets which are defined in the VPD table are displayed in full mode of the SCOS-2000 Packets History Displays.

Add remark about the use of the PID table for internally generated TM packets (e.g. UDC, MISC and SPPG packets).

DCR-263: state that the fields OCP_RLCHK and OCP_VALPAR (checks applicability criteria) are not used for Status Consistency Checks.

Clarify the usage of the field OCF_NBCHCK when invalid parameter values are received.

Clarify the start reference for the position fields in the PIC table (replace "end of SCOS-2000 TM header" with "start of the TM Source Packet Header").

Commanding

DCR-255: Increase the allowed number of digits for the field CSF_ELEMS from 3 to 5.

Clarify constraint in the ability to associate an interlock to commands belonging to a Block/Group (field CSS_ILSCOPE).

Clarify how to populate the bit offset in the CDF table for elements following a repeated group.

Clarify that the CDF definition of the 'Insert Telecommand in the Command Schedule' TC is not used (only its CCF definition)

Clarify how to populate the field CCF_ACK.



DCR-188: state that the inclusion of High-priority commands into a block is not allowed. Clarify the usage of blocking by the S2K releaser (packets are aggregated into a single segment).

DCR-241: introduce a new category of commands (field CCF_TYPE="S") for which no CRC is appended by the Releaser.

DCR-242: add a new optional table identifying the routing and destination of commands (required by EGSE systems only).

DCR-266: Change required for EGSE system: The CCF_APID, CCF_RAPID and PID_APID length has to be changed from num(4) to num(5). 0 to 2047 is used to identify an onboard APID. 2048 to 65535 is used to identify a ground process id.