

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

## SatView™ Database Standards

All information is subject to change without notice and does not represent a commitment on the part of **BINARY SPACE**.  
Release 2.01 (July 2015)

### Table of Contents

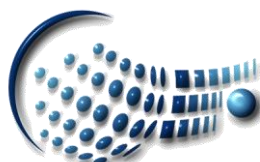
|                               |          |
|-------------------------------|----------|
| <b>1. Introduction</b>        | <b>3</b> |
| <b>2. Architecture</b>        | <b>3</b> |
| <b>3. Supported Databases</b> | <b>4</b> |
| 3.1. ALTEL Standard           | 4        |
| 3.2. SCOS-2000 Standard       | 31       |

### Table of Figures

| Figure | Description                 |
|--------|-----------------------------|
| 2.1.   | SatView™ Exposed Interfaces |

### Document Change Log

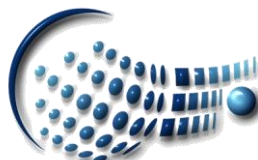
| Issue | Revision | Date                           | Affected  | Reason for change                     |
|-------|----------|--------------------------------|---|---------------------------------------|
| 1     | 1        | July 1998                      | All   | New document                          |
| 1     | 2        | October 2001                   | 3.1.18.<br>3.1.19.                                  | Alphanumeric Displays (ANDs)          |
| 1     | 3        | December 2001                  | 3.1.22.   | Parameter Observation Displays (PODs) |
| 1     | 4        | May 2002                       | 3.1.20.<br>3.1.21.                                  | Graphic Displays (GRDs)               |
| 1     | 5        | January 2004<br><br>April 2004 | 3.1.13.<br>3.1.16.<br>3.1.18.<br>3.1.20.<br>3.1.22. | Record change status field added      |
| 1     | 6        | October 2004                   | 3.1.1.  | Deleted spacecraft field              |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|   |      |            |  |  |
|---|------|------------|--|--|
| 1 | 7    | April 2006 | 3.1.4.   | Telemetry Packet Streams   |
| 1 | 8-10 | June 2006  | 3.1.23.<br>3.1.24.<br>3.1.25.<br>3.1.26.<br>3.1.27.<br>3.1.28.<br>3.1.29.<br>3.1.30.<br>3.1.31.<br>3.1.32.<br>3.1.33.<br>3.1.34.<br>3.1.35.<br>3.1.36.<br>3.1.37.<br>3.1.38.<br>3.1.39.<br>3.1.40.<br>3.1.41.<br>3.1.42.<br>3.1.43.<br>3.1.44.<br>3.1.45.<br>3.1.46. | Telecommand Functions<br><br>Telecommand Packets<br><br>Telecommand Parameters<br>Telecommand Parameter Sets<br><br>Pre-execution Verification Checks<br><br>Execution Verification Checks<br><br>Telecommand Sequences<br><br><br>Telecommand Blocks<br><br>On-Board Processors |
| 2 | 1    | July 2015  | 3.2.   | SCOS-2000 Database Standard  |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

### 1. Introduction

This document provides a description of the database standards supported by SatView™.

### 2. Architecture

SatView™ is designed to support a variety of database standards. This flexibility is achieved by special software drivers encapsulating all database related tasks (e.g. DBMS interface, database parsing, data visualization etc.).

The database interface is one of several interfaces exposed by SatView™ to third-party developers allowing for an adaption to a custom infrastructure:

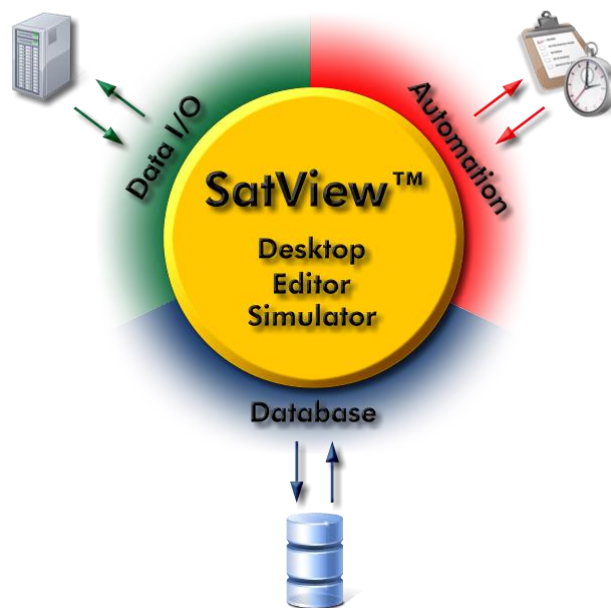
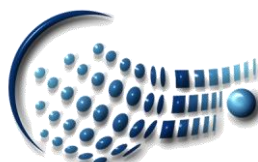


Figure 2.1. - SatView™ Exposed Interfaces

Consult the related document 'SatView™ Database Interface' for more details about how to implement a database driver.



### 3. Supported Databases

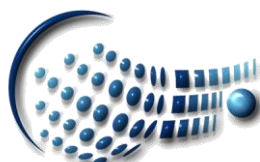
Currently, the following database standards are supported by SatView™:

| Name      | Owner                       | Spacecraft | DBMS                  |
|-----------|-----------------------------|------------|-----------------------|
| ALTEL     | Level 3 Communications Inc. | ARTEMIS    | Microsoft® SQL Server |
| SCOS-2000 | ESA                         |            | Microsoft® SQL Server |

#### 3.1. ALTEL Standard

The following chapters describe the tables and business rules used for the ALTEL database:

| Chapter | Table | Description   | Page |
|---------|-------|---|------|
| 3.1.1.  | PMF   | Defines all telemetry packets generated by the subsystems and payload instruments | 6    |
| 3.1.2.  | DFH   | Defines all distinct telemetry packet data field headers                          | 7    |
| 3.1.3.  | PLF   | Defines all parameter locations within telemetry packets                          | 7    |
| 3.1.4.  | STR   | Defines all telemetry packet streams  | 8    |
| 3.1.5.  | PCF   | Defines all telemetry parameters  | 8    |
| 3.1.6.  | PCAL  | Associates numerical calibration curves with telemetry parameters                 | 10   |
| 3.1.7.  | CAL   | Defines all numerical calibration curves  | 10   |
| 3.1.8.  | CALP  | Defines all points related to the numerical calibration curves                    | 10   |
| 3.1.9.  | PTXF  | Associates status calibration tables with telemetry parameters                    | 11   |
| 3.1.10. | TXF   | Defines all status calibration curves   | 11   |
| 3.1.11. | TXFP  | Defines all translations related to the status calibration curves                 | 11   |
| 3.1.12. | POCF  | Associates numerical out-of-limit checks with telemetry parameters                | 11   |
| 3.1.13. | OCF   | Defines all numerical out-of-limit check tables                                   | 12   |
| 3.1.14. | OCFP  | Defines all numerical checks of the out-of-limit tables                           | 12   |
| 3.1.15. | PFXF  | Associates status out-of-limit checks with telemetry parameters                   | 12   |
| 3.1.16. | FXF   | Defines all status out-of-limit check tables                                      | 13   |
| 3.1.17. | FXFP  | Defines all status checks of the out-of-limit tables                              | 13   |
| 3.1.18. | DPF   | Defines all alphanumeric displays (ANDs)  | 13   |
| 3.1.19. | DPFP  | Defines all parameters contained in the alphanumeric displays                     | 14   |
| 3.1.20. | GPF   | Defines all graphic displays (GRDs)   | 15   |
| 3.1.21. | GPFP  | Defines all parameters contained in the graphic displays                          | 16   |
| 3.1.22. | PPF   | Defines all parameter observation displays (PODs)                                 | 17   |
| 3.1.23. | CCF   | Defines all telecommand functions   | 18   |
| 3.1.24. | CDF   | Defines all parameters contained in the telecommand functions                     | 19   |
| 3.1.25. | TCP   | Defines all telecommand packets   | 19   |
| 3.1.26. | PCDF  | Defines all parameters related to the telecommand packets                         | 20   |
| 3.1.27. | CPC   | Defines all telecommand parameters  | 21   |
| 3.1.28. | PCPS  | Associates parameter sets with telecommand functions                              | 21   |
| 3.1.29. | CPS   | Defines all telecommand parameter sets  | 21   |





# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|         |      |   |    |
|---------|------|---|----|
| 3.1.30. | CPSP | Defines all parameters of the telecommand parameter sets                              | 22 |
| 3.1.31. | PVP  | Defines all pre-execution verification parameter groups                               | 22 |
| 3.1.32. | PEV  | Defines all parameters contained in the pre-execution verification parameter groups   | 22 |
| 3.1.33. | CVP  | Defines all execution verification parameter groups                                   | 23 |
| 3.1.34. | CEV  | Defines all parameters contained in the execution verification parameter groups       | 23 |
| 3.1.35. | CSF  | Defines all telecommand sequences   | 24 |
| 3.1.36. | CSS  | Defines all functions, sequences or procedures contained in the telecommand sequences | 24 |
| 3.1.37. | CSP  | Defines all formal parameters related to the telecommand sequences                    | 25 |
| 3.1.38. | SDF  | Defines all parameters contained in the telecommand sequence entries                  | 26 |
| 3.1.39. | BCF  | Defines all telecommand blocks  | 27 |
| 3.1.40. | BCCF | Defines all functions contained in the telecommand blocks                             | 27 |
| 3.1.41. | GEN  | Defines all on-board processors   | 27 |
| 3.1.42. | MAD  | Defines all memory areas contained in the on-board processors                         | 28 |
| 3.1.43. | MEM  | Defines all memory blocks contained in the memory areas                               | 28 |
| 3.1.44. | IMG  | Defines all memory images contained in the memory areas                               | 29 |
| 3.1.45. | CRC  | Defines all check sums attached to the memory areas                                   | 29 |
| 3.1.46. | CAT  | Defines all catalogued memory area versions   | 29 |



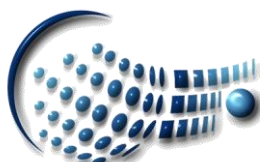
# BINARY SPACE

## RELIABLE SPACE SYSTEMS

### 3.1.1. PMF Table

This table provides the complete list of telemetry packets generated by the on-board subsystems and payload instruments.

| No. | Column       | Type    | Description  | Restriction  |
|-----|--------------|---------|--|--|
| 1   | PMF_PACKID   | ch(4)   | Telemetry packet name  | Unique, not null   |
| 2   | PMF_TYPE     | ch(1)   | Telemetry packet type  | Must be 0, not null  |
| 3   | PMF_DESCR    | vc(32)  | Telemetry packet description   | None, not null   |
| 4   | PMF_HFLG     | int     | Data field header flag:<br>0=No header<br>1=Header exists                                    | In set {0,1}   |
| 5   | PMF_APID     | ch(3)   | Application identifier (hex)   | In set {000,001,002,003,004,011,012,013,020,021,022}<br>Not null   |
| 6   | PMF LENG     | int     | Packet data field size (in bytes) minus 1  | 0-121<br>Not null  |
| 7   | PMF_TMTY     | ch(1)   | Telemetry type:<br>0=Cyclic<br>1=Exception<br>2=Oversampled<br>3=Report<br>4=Dump            | In set {0,1,2,3,4} or null   |
| 8   | PMF TMID     | ch(2)   | Telemetry identifier (hex)   | 00-FF or null  |
| 9   | PMF_FUNC     | vc(254) | Extended packet description  | None, null   |
| 10  | PMF_SMPRT    | int     | Packet transmission rate (in seconds)  | >=0 or null  |
| 11  | PMF_VALID    | ch(6)   | Number of I/O failures before the packet is set to invalid (only used for AOCS oversampling) | >0 or null   |
| 12  | PMF_VARIABLE | int     | Variable packet length flag:<br>0=Fixed length<br>1=Variable length                          | In set {0,1} or null   |
| 13  | PMF_3VAL     | vc(4)   | Third discriminator  | 0000-FFFF or null  |
| 14  | PMF_STATUS   | ch(1)   | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted                          | In set {N,T,C,D} or null   |
| 15  | PMF_DFH_NAME | vc(12)  | Data field header name   | If PMF_HFLG=1 then must be a valid data field header name<br>If PMF_HFLG=0 then must be null                                   |
| 16  | PMF_VCID     | int     | Virtual channel identifier   | In set {0,1}   |
| 17  | PMF_EVENTID  | ch(6)   | Name of the event identifier parameter   | Must be a valid parameter name located in this packet or a DS16 parameter whose parent is located there as well<br>May be null |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

### 3.1.2. DFH Table

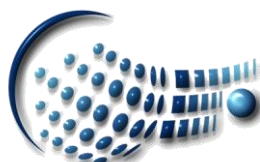
This table provides the complete list of all distinct telemetry packet data field headers.

| No. | Column                 | Type   | Description  | Restriction  |
|-----|------------------------|--------|--|--|
| 1   | DFH_NAME               | vc(12) | Data field header name   | Unique, not null   |
| 2   | DFH_TM_TYPE_PRESENT    | int    | Telemetry type field flag:<br>0 = No telemetry type field<br>1 = Telemetry type field exists                   | In set {0,1}   |
| 3   | DFH_TM_TYPE_BIT_OFFSET | int    | Bit offset of the telemetry type field from the start of the data field header                                 | If DFH_TM_TYPE_PRESENT=1 then $\geq 0$<br>If DFH_TM_TYPE_PRESENT=0 then null |
| 4   | DFH_TM_TYPE_BIT_LENGTH | int    | Length of the telemetry type field (in bits)   | If DFH_TM_TYPE_PRESENT=1 then $\geq 1$<br>If DFH_TM_TYPE_PRESENT=0 then null |
| 5   | DFH_TM_ID_PRESENT      | int    | Telemetry identifier field flag:<br>0 = No telemetry identifier field<br>1 = Telemetry identifier field exists | In set {0,1}   |
| 6   | DFH_TM_ID_BIT_OFFSET   | int    | Bit offset of the telemetry identifier field from the start of the data field header                           | If DFH_TM_ID_PRESENT=1 then $\geq 0$<br>If DFH_TM_ID_PRESENT=0 then null     |
| 7   | DFH_TM_ID_BIT_LENGTH   | int    | Length of the telemetry identifier field (in bits)   | If DFH_TM_ID_PRESENT=1 then $\geq 1$<br>If DFH_TM_ID_PRESENT=0 then null     |
| 8   | DFH_3DISC_PRESENT      | int    | Third descriptor flag:<br>0 = No third descriptor field<br>1 = Third descriptor field exists                   | In set {0,1}   |
| 9   | DFH_3DISC_BIT_OFFSET   | int    | Bit offset of the third descriptor field from the start of the data field header                               | If DFH_3DISC_PRESENT=1 then $\geq 0$<br>If DFH_3DISC_PRESENT=0 then null     |
| 10  | DFH_3DISC_BIT_LENGTH   | int    | Length of the third descriptor field (in bits)   | If DFH_3DISC_PRESENT=1 then $\geq 1$<br>If DFH_3DISC_PRESENT=0 then null     |
| 11  | DFH_TIME_PRESENT       | int    | Time stamp flag:<br>0 = No time stamp<br>1 = Time stamp exists   | In set {0,1}   |
| 12  | DFH_TIME_BIT_OFFSET    | int    | Bit offset of the timestamp from the start of the data field header  | If DFH_TIME_PRESENT=1 then $\geq 0$<br>If DFH_TIME_PRESENT=0 then null       |
| 13  | DFH_TIME_BIT_LENGTH    | int    | Length of the time stamp (in bits)   | If DFH_TIME_PRESENT=1 then $\geq 1$<br>If DFH_TIME_PRESENT=0 then null       |

### 3.1.3. PLF Table

This table provides the complete list of all parameter locations within telemetry packets.

| No. | Column     | Type  | Description  | Restriction                      |
|-----|------------|-------|--|----------------------------------|
| 1   | PLF_NAME   | ch(6) | Telemetry parameter name                                     | Valid parameter name<br>Not null |
| 2   | PLF_PACKID | ch(4) | Telemetry packet name  | Valid packet name<br>Not null    |
| 3   | PLF_OFFBY  | int   | Location of the first occurrence of the parameter (in bytes) | 0-121<br>Not null                |
| 4   | PLF_OFFBI  | int   | Location of the first bit within the specified byte          | 0-7<br>Not null                  |
| 5   | PLF_NBOCC  | int   | Number of occurrences  | 1-976<br>Not null                |
| 6   | PLF_NBGRP  | int   | Number of groups   | In set {1}                       |
| 7   | PLF_LGOCC  | int   | Number of bits between successive occurrences                | 0-974 or null                    |
| 8   | PLF_LGGRP  | int   | Not used   | Not applicable and may be null   |
| 9   | PLF_TIME   | int   | Not used   | Not applicable and may be null   |
| 10  | PLF_TDOCC  | int   | Not used   | Not applicable and may be null   |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |            |       |  |   |
|----|------------|-------|--|---|
| 11 | PLF_VALID  | ch(6) | Validity parameter                                       | Must refer to a parameter with a width of 1 bit or null |
| 12 | PLF_STATUS | ch(1) | Record change status:<br>N=New<br>C=Changed<br>D=Deleted | In set {N,C,D} or null                                  |

### 3.1.4. STR Table

This table provides the complete list of all telemetry packet streams.

| No. | Column      | Type  | Description                                  | Restriction   |
|-----|-------------|-------|--|---|
| 1   | STR_STID    | ch(6) | Stream name                                  | Unique, not null  |
| 2   | STR_APID    | ch(3) | Application identifier this stream refers to | Valid application identifier<br>Not null                |
| 3   | STR_TMID    | ch(2) | Telemetry identifier                         | Valid telemetry identifier or null                      |
| 4   | STR_PERIOD  | int   | Stream period (in milliseconds)              | >=0, not null   |
| 5   | STR_TIMEOUT | int   | Stream timeout (in milliseconds)             | >=0, not null   |
| 6   | STR_VALID   | ch(6) | Validity parameter                           | Must refer to a parameter with a width of 1 bit or null |

### 3.1.5. PCF Table

This table provides the complete list of telemetry parameters.

| No. | Column     | Type   | Description  | Restriction  |
|-----|------------|--------|--|--|
| 1   | PCF_NAME   | ch(6)  | Telemetry parameter name   | Unique, not null   |
| 2   | PCF_DESCR  | vc(25) | Telemetry parameter description  | None, null   |
| 3   | PCF_UNIT   | vc(4)  | Telemetry parameter unit   | None, null   |
| 4   | PCF_VALID  | ch(6)  | Validity parameter   | Must refer to a parameter with a width of 1 bit or null  |
| 5   | PCF_CATEG  | ch(1)  | Parameter calibration:<br>N=Numerical<br>S=Status<br>R=None  | In set {N,S,R}<br>Not null   |
| 6   | PCF_WIDTH  | int    | Parameter width (in bits)  | If PCF_NATURE=C then ignored<br>If PCF_NATURE=D then 1 or 32<br>If PCF_NATURE=R then >0 and <=32<br>Not null |
| 7   | PCF_NATURE | ch(1)  | Nature of parameter:<br>R=Raw<br>D=Derived<br>C=Constant   | In set {R,D,C} and not null<br>If PCF_NATURE=D then PCF_DRVEXP must hold an expression                       |
| 8   | PCF_CODING | ch(1)  | Type of digital coding:<br>B=Bit pattern<br>I=Signed integer<br>U=Unsigned integer<br>S=Complement complex<br>1=Floating point | If PCF_NATURE<>C then in set {B,I,U,S,1}<br>If PCF_NATURE=C then may be null                                 |
| 9   | PCF_INTER  | ch(1)  | Type of interpretation:<br>E=Perform an extrapolation<br>F=No extrapolation  | If PCF_CATEG=N then in set {E,F}<br>If PCF_CATEG<>N then may be null   |
| 10  | PCF_USECON | ch(1)  | Consistency check flag:<br>Y=Enabled<br>N=Disabled   | In set {Y,N} or null   |

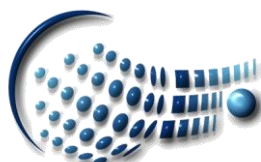




# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |            |         |  |  |
|----|------------|---------|--|--|
| 11 | PCF_PARVAL | vc(12)  | Constant raw parameter value   | If PCF_NATURE=C then<br>Decimal number $n$ : $n$<br>Hexadecimal number $n$ : $\#n$<br>Text string: $'text'$<br>If PCF_NATURE<>C then ignored and may be null   |
| 12 | PCF_TYPE   | vc(5)   | Parameter type:<br>DS-16=Serial Digital (16 bits)<br>DS-8 =Serial Digital (8 bits)<br>BI-L =Bilevel<br>HBI-L =High priority bilevel<br>AN =Analog              | In set {DS-16,DS-8,BI-L,HBI-L,AN} and not null   |
| 13 | PCF_LDESC  | vc(48)  | Parameter description  | None, null   |
| 14 | PCF_SOURS  | vc(5)   | Source subsystem of the parameter  | None, null   |
| 15 | PCF_SOURU  | vc(5)   | Source equipment of the parameter  | None, null   |
| 16 | PCF_ROUTS  | vc(5)   | Subsystem by which the parameter is routed to OBCU   | None, null   |
| 17 | PCF_ROUTU  | vc(5)   | Equipment by which the parameter is routed to OBCU   | None, null   |
| 18 | PCF_ASTC   | vc(128) | Tag numbers of up to 8 associated telecommands   | None, null   |
| 19 | PCF_OPINFO | vc(254) | Text description of operational information  | None, null   |
| 20 | PCF_APPLC  | vc(254) | Textual description of the conditions that must be true for the parameter to be valid  | None, null   |
| 21 | PCF_MONIT  | vc(254) | Textual description of the warning and danger limits   | None, null   |
| 22 | PCF_ALTM   | vc(39)  | Up to 8 alternative telemetry parameters. If this parameter is a sub-parameter of a DS-16 channel then the first entry is the parent parameter of the channel. | Parameters delimited by commas or null   |
| 23 | PCF_INTM   | vc(6)   | OBDAH interrogation message instruction  | Hexadecimal number between 0 and FFFFFFF or null   |
| 24 | PCF_DRVEXP | text    | Derivation expression  | If PCF_NATURE=D then valid expression<br>If PCF_NATURE<>D then null  |
| 25 | PCF_EXPTRG | vc(20)  | Trigger parameter  | If PCF_NATURE=D then<br>PCF_TRGTYP   PCF_EXPTRG<br>C   Null<br>K   Valid packet name<br>P   Valid parameter name or xxx_yyy where xxx refers to a valid parameter and yyy to a valid packet<br>If PCF_NATURE<>D then ignored |
| 26 | PCF_TRGTYP | ch(1)   | Trigger type:<br>C=Constant<br>K=Packet<br>P=Parameter   | If PCF_NATURE=D then in set {C,K,P}<br>If PCF_NATURE<>D then null  |
| 27 | PCF_OBSPAR | ch(1)   | On-board flag:<br>S=On-board software parameter<br>N=Monitoring parameter<br>B=Both  | If PCF_NATURE=R then in set {S,N,B}<br>If PCF_NATURE<>R then ignored   |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |            |       |   |                          |
|----|------------|-------|---|--------------------------|
| 28 | PCF_STATUS | ch(1) | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null |
|----|------------|-------|---|--------------------------|

### 3.1.6. PCAL Table

This table provides the complete list of all associations between the numerical calibration curves and telemetry parameters.

| No. | Column      | Type  | Description  | Restriction  |
|-----|-------------|-------|--|--|
| 1   | PCAL_PNAME  | ch(6) | Telemetry parameter name                                 | Valid numerical parameter name<br>Not null         |
| 2   | PCAL_CNAME  | vc(8) | Calibration curve name                                   | Valid numerical calibration curve name<br>Not null |
| 3   | PCAL_VALID  | ch(6) | Validity parameter                                       | Valid parameter name or null                       |
| 4   | PCAL_STATUS | ch(1) | Record change status:<br>N=New<br>C=Changed<br>D=Deleted | In set {N,C,D} or null                             |

### 3.1.7. CAL Table

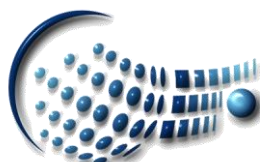
This table provides the complete list of all numerical calibration curves.

| No. | Column     | Type   | Description   | Restriction              |
|-----|------------|--------|---|--------------------------|
| 1   | CAL_NAME   | vc(8)  | Calibration curve name  | Unique, not null         |
| 2   | CAL_DESCR  | vc(32) | Calibration curve description                                       | None, not null           |
| 3   | CAL_UNIT   | vc(12) | Calibration curve units   | None                     |
| 4   | CAL_NBPNT  | int    | Number of points in curve   | >=0, not null            |
| 5   | CAL_STATUS | ch(1)  | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null |

### 3.1.8. CALP Table

This table provides the complete list of all points related to the numerical calibration curves.

| No. | Column      | Type  | Description  | Restriction  |
|-----|-------------|-------|--|--|
| 1   | CALP_NAME   | vc(8) | Calibration curve name                                   | Valid numerical calibration curve name<br>Not null |
| 2   | CALP_XVAL   | float | X value of point on curve                                | Valid floating point number<br>Not null            |
| 3   | CALP_YVAL   | float | Y value of point on curve                                | Valid floating point number<br>Not null            |
| 4   | CALP_STATUS | ch(1) | Record change status:<br>N=New<br>C=Changed<br>D=Deleted | In set {N,C,D} or null                             |



### 3.1.9. PTXF Table

This table provides the complete list of all associations between the status calibration tables and telemetry parameters.

| No. | Column      | Type  | Description  | Restriction                                     |
|-----|-------------|-------|--|---|
| 1   | PTXF_PNAME  | ch(6) | Telemetry parameter name                                 | Valid status parameter name<br>Not null         |
| 2   | PTXF_CNAME  | vc(8) | Calibration curve name                                   | Valid status calibration curve name<br>Not null |
| 3   | PTXF_VALID  | ch(6) | Validity parameter                                       | Valid parameter name or null                    |
| 4   | PTXF_STATUS | ch(1) | Record change status:<br>N=New<br>C=Changed<br>D=Deleted | In set {N,C,D} or null                          |

### 3.1.10. TXF Table

This table provides the complete list of all status calibration curves.

| No. | Column     | Type   | Description   | Restriction              |
|-----|------------|--------|---|--------------------------|
| 1   | TXF_NAME   | vc(8)  | Calibration curve name  | Unique, not null         |
| 2   | TXF_DESCR  | vc(32) | Calibration curve description                                       | None, not null           |
| 3   | TXF_NBTXT  | int    | Number of text strings in the curve                                 | 1-255<br>Not null        |
| 4   | TXF_STATUS | ch(1)  | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null |

### 3.1.11. TXFP Table

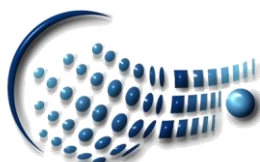
This table provides the complete list of all translations related to the status calibration curves.

| No. | Column      | Type   | Description  | Restriction                                     |
|-----|-------------|--------|--|---|
| 1   | TXFP_NAME   | vc(8)  | Calibration curve name                                   | Valid status calibration curve name<br>Not null |
| 2   | TXFP_FROM   | int    | Raw value (low limit of range)                           | Valid number, not null                          |
| 3   | TXFP_TO     | int    | Raw value (high limit of range)                          | Valid number, not null                          |
| 4   | TXFP_TEXT   | vc(12) | Status text  | None, not null                                  |
| 5   | TXFP_STATUS | ch(1)  | Record change status:<br>N=New<br>C=Changed<br>D=Deleted | In set {N,C,D} or null                          |

### 3.1.12. POCF Table

This table provides the complete list of all associations between numerical out-of-limit checks and telemetry parameters.

| No. | Column     | Type   | Description              | Restriction                                       |
|-----|------------|--------|--------------------------|---|
| 1   | POCF_PNAME | ch(6)  | Telemetry parameter name | Valid numerical parameter name<br>Not null        |
| 2   | POCF_ONAME | vc(10) | Out-of-limit set name    | Valid numerical out-of-limit set name<br>Not null |
| 3   | POCF_VALID | ch(6)  | Validity parameter       | Valid parameter name or null                      |



### 3.1.13. OCF Table

This table provides the complete list of all numerical out-of-limit check tables.

| No. | Column      | Type   | Description  | Restriction  |
|-----|-------------|--------|--|--|
| 1   | OCF_ONAME   | vc(10) | Numerical out-of-limit set name  | Unique, not null   |
| 2   | OCF_NBCHECK | int    | Number of times the check must fail before considered as out-of-limit condition  | If OCF_INTER in set {C,U} then 1-9<br>If OCF_INTER in set {E,R} then null              |
| 3   | OCF_INTER   | ch(1)  | Limits interpretation:<br>C=Calibrated TM check<br>U=Raw (uncalibrated) TM check<br>E=Calibrated TC check<br>R=Raw (uncalibrated) TC check | In set {C,U,E,R}   |
| 4   | OCF_CODING  | ch(1)  | Limits coding:<br>U=Unsigned integer<br>I =Signed integer<br>1 =Floating point   | If OCF_INTER in set {U,R} then in set {U,I,1}<br>If OCF_INTER in set {C,E} then null   |
| 5   | OCF_RADIX   | ch(1)  | Limits radix:<br>H=Hexadecimal<br>D=Decimal<br>B =Binary<br>O=Octal  | If OCF_INTER in set {U,R} then in set {H,D,B,O}<br>If OCF_INTER in set {C,E} then null |
| 6   | OCF_STATUS  | ch(1)  | Record change status:<br>N=New<br>T =Draft<br>C=Changed<br>D=Deleted   | In set {N,T,C,D} or null   |

### 3.1.14. OCFP Table

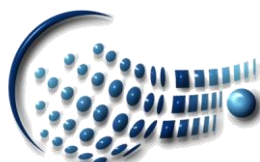
This table provides the complete list of all numerical checks related to the out-of-limit tables.

| No. | Column      | Type   | Description   | Restriction   |
|-----|-------------|--------|---|---|
| 1   | OCFP_ONAME  | vc(10) | Numerical out-of-limit set name                                       | Valid out-of-limit set name<br>Not null                       |
| 2   | OCFP_ORDER  | int    | Out-of-limit check order:<br>1=Soft limit check<br>2=Hard limit check | In set {1,2}  |
| 3   | OCFP_LVALUE | vc(32) | Low value of check range  | Valid numerical value<br>OCFP_LVALUE<=OCFP_HVALUE<br>Not null |
| 4   | OCFP_HVALUE | vc(32) | High value of check range   | Valid numerical value<br>OCFP_LVALUE<=OCFP_HVALUE<br>Not null |

### 3.1.15. PFXF Table

This table provides the complete list of all associations between status out-of-limit checks and telemetry parameters.

| No. | Column     | Type   | Description              | Restriction                                    |
|-----|------------|--------|--------------------------|--|
| 1   | PFXF_PNAME | ch(6)  | Telemetry parameter name | Valid status parameter name<br>Not null        |
| 2   | PFXF_SNAME | vc(10) | Out-of-limit set name    | Valid status out-of-limit set name<br>Not null |
| 3   | PFXF_VALID | ch(6)  | Validity parameter       | Valid parameter name or null                   |





### 3.1.16. FXF Table

This table provides the complete list of all status out-of-limit check tables.

| No. | Column      | Type   | Description  | Restriction  |
|-----|-------------|--------|--|--|
| 1   | FXF_SNAME   | vc(10) | Status out-of-limit set name   | Unique, not null   |
| 2   | FXF_NBCHECK | int    | Number of times the check must fail before considered as out-of-limit condition  | 1-99<br>Not null   |
| 3   | FXF_INTER   | ch(1)  | Limits interpretation:<br>C=Calibrated TM check<br>U=Raw (uncalibrated) TM check<br>E=Calibrated TC check<br>R=Raw (uncalibrated) TC check | In set {C,U,E,R}   |
| 4   | FXF_RADIX   | ch(1)  | Limits radix:<br>H=Hexadecimal<br>D=Decimal<br>B=Binary<br>O=Octal   | If FXF_INTER in set {U,R} then in set {H,D,B,O}<br>If FXF_INTER in set {C,E} then null |
| 5   | FXF_STATUS  | ch(1)  | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted  | In set {N,T,C,D} or null   |

### 3.1.17. FXFP Table

This table provides the complete list of all status checks related to the out-of-limit tables.

| No. | Column     | Type   | Description                  | Restriction  |
|-----|------------|--------|------------------------------|--|
| 1   | FXFP_SNAME | vc(10) | Status out-of-limit set name | Valid out-of-limit set name<br>Not null  |
| 2   | FXFP_ORDER | int    | Out-of-limit check order     | > 0, unique, not null  |
| 3   | FXFP_FROM  | vc(12) | Low value of check range     | Valid number of text<br>If FXF_INTER in set {U,R} then FXFP_FROM <= FXFP_TO (non-overlapped)<br>Not null |
| 4   | FXFP_TO    | vc(12) | High value of check range    | Valid number of text<br>If FXF_INTER in set {U,R} then FXFP_FROM <= FXFP_TO (non-overlapped)<br>Not null |

### 3.1.18. DPF Table

This table provides the complete list of alphanumeric displays (AND). It is an extension to the database format delivered by ALTEL.

| No. | Column     | Type    | Description  | Restriction  |
|-----|------------|---------|--|--|
| 1   | DPF_NUMBER | ch(6)   | Name of the AND  | Unique, A-Z, a-z or 0-9<br>Not null                                    |
| 2   | DPF_TITLE  | vc(32)  | Title  | None, not null   |
| 3   | DPF_LAYOUT | vc(256) | Column layout: $n_0, n_1, n_2, \dots$<br>Layout code $n_i$ :<br>ID= $2^0$   Description= $2^1$  <br>Value= $2^2$   Unit= $2^3$   Change TM<br>Unit= $2^4$   Change TM Time= $2^5$  <br>Update TM Unit= $2^6$   Update TM<br>Time= $2^7$   Expected Values= $2^8$ | $1 \leq n_i < 512$ or -1 (default)<br>Modulo( $n_i, 2$ )=1<br>Not null |



# BINARY SPACE

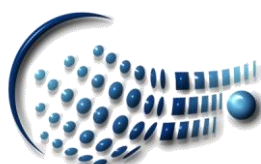
## RELIABLE SPACE SYSTEMS

|   |            |       |  |                          |
|---|------------|-------|--|--------------------------|
| 4 | DPF_STATUS | ch(1) | Record change status:<br>N=New<br>T =Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null |
|---|------------|-------|--|--------------------------|

### 3.1.19. DPFP Table

This table provides the complete list of parameters contained in the alphanumeric displays. It is an extension to the database format delivered by ALTEL.

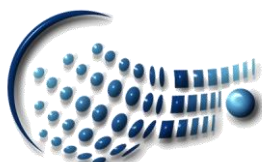
| No. | Column        | Type   | Description   | Restriction  |
|-----|---------------|--------|---|--|
| 1   | DPFP_NUMBER   | ch(6)  | Name of the AND   | Unique, A-Z, a-z or 0-9<br>Not null  |
| 2   | DPFP_PNAME    | ch(6)  | Parameter name  | Valid parameter name or empty (group label)<br>Not null  |
| 3   | DPFP_COMMENT  | vc(32) | Comment related to the parameter, group label and/or expected values (enclosed by <Expected Values>)  | None, not null (group label)   |
| 4   | DPFP_ROW      | int    | Row position of the parameter or of the group label   | >=0, <DPFP_ROW,DPFP_COL> unique<br>Not null  |
| 5   | DPFP_COL      | int    | Column position of the parameter or of the group label  | >=0, <DPFP_ROW,DPFP_COL> unique<br>Not null  |
| 6   | DPFP_COLOR    | int    | Color of the parameter or of the group label  | Valid RGB color: <Red,Green,Blue><br>0...2 <sup>24</sup> -1 or -1 (default)<br>Not null                            |
| 7   | DPFP_COMM     | int    | Occurrence or past sample of the parameter  | >=0: Occurrence number within the telemetry format<br><0: Past sample number<br>Use 0 for group labels<br>Not null |
| 8   | DPFP_VALMODE  | ch(1)  | Display mode of the parameter value:<br>0 =Default (calibrated)<br>R =Raw (coded)<br>B =Binary (raw, not coded)<br>I =Inverse binary (as above)<br>O=Octal (as above)<br>D=Decimal (as above)<br>H=Hexadecimal (as above) | In set {0,R,B,I,O,D,H}<br>Use 0 for group labels<br>Not null   |
| 9   | DPFP_VALCHARS | int    | Number of character positions for the parameter value   | >0 or 0 (default)<br>Use 0 for group labels<br>Not null  |
| 10  | DPFP_VALALIGN | ch(1)  | Alignment of the parameter value or of the group label:<br>L =Left<br>C=Center<br>R=Right   | In set {L,C,R}<br>Not null   |



### 3.1.20. GPF Table

This table provides the complete list of graphic displays (GRD). It is an extension to the database format delivered by ALTEL.

| No. | Column       | Type   | Description  | Restriction   |
|-----|--------------|--------|--|---|
| 1   | GPF_NUMBER   | ch(6)  | Name of the GRD  | Unique, A-Z, a-z or 0-9<br>Not null   |
| 2   | GPF_TITLE    | vc(32) | Title  | None, not null  |
| 3   | GPF_TYPE     | ch(1)  | Plot type:<br>0=Normal (Parameter-to-time)<br>1=Stripchart (Parameter-to-time, channeled)<br>2=Stackchart (Parameter-to-time, stacked)<br>3=Diagram (Parameter-to-parameter)                   | In set {0,1,2,3}<br>Not null  |
| 4   | GPF_SUBTYPE  | ch(1)  | Plot subtype:<br>0=Normal (lines only)<br>1=Area (colored space between lines)   | In set {0,1}<br>Must be 0 for GPF_TYPE=3<br>Not null                                    |
| 5   | GPF_MODE     | ch(1)  | Plot mode:<br>0=Scroll<br>1=Overwrite<br>2=Restart   | In set {0,1,2}<br>Must be 2 for GPF_TYPE=3<br>Not null                                  |
| 6   | GPF_LAYOUT   | int    | Column layout $n$ :<br>ID= $2^0$   Description= $2^1$  <br>Value= $2^2$   Unit= $2^3$   Change TM<br>Unit= $2^4$   Change TM Time= $2^5$  <br>Update TM Unit= $2^6$   Update TM<br>Time= $2^7$ | $1 < n < 1024$ or -1 (default)<br>Modulo( $n,2$ )=1<br>Not null                         |
| 7   | GPF_DAYS     | int    | Duration (in days)   | $0 \leq \text{Days} < 100$ ,<br>Duration > 0<br>Not null                                |
| 8   | GPF_HOURS    | int    | Duration (in hours)  | $0 \leq \text{Hours} < 24$ ,<br>Duration > 0<br>Not null                                |
| 9   | GPF_MINUTES  | int    | Duration (in minutes)  | $0 \leq \text{Minutes} < 60$ , Duration > 0<br>Not null                                 |
| 10  | GPF_INTERVAL | int    | Visible interval (in minutes)  | $0 < \text{Interval} \leq \text{Duration}$<br>Must be 0 for GPF_MODE=1 or 2<br>Not null |
| 11  | GPF_UPDATE   | int    | Update interval (in minutes):<br>$n > 0$ : Refreshes the plots every $n$ minutes<br>$n = 0$ : Refreshes the plots as fast as possible  | $0 < \text{Interval} \leq \min(1440, \text{Duration})$ or<br>0 (default)<br>Not null    |
| 12  | GPF_WIDTH    | int    | Plot area width (in pixels)  | >0 or 0 (default)<br>Not null   |
| 13  | GPF_HEIGHT   | int    | Plot area height (in pixels)   | >0 or 0 (default)<br>Not null   |
| 14  | GPF_XGRIDS   | int    | Grids on the horizontal plot area axis   | 0-100<br>Not Null   |



# BINARY SPACE

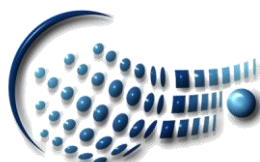
## RELIABLE SPACE SYSTEMS

|    |                |       |   |   |
|----|----------------|-------|---|---|
| 15 | GPF_YGRIDS     | int   | Grids on the vertical plot area axis                                | 0-100<br>Must be 0 for GPF_TYPE=1<br>Not null   |
| 16 | GPF_XTICKS     | int   | Ticks on the horizontal plot area axis                              | 0-100<br>Not null   |
| 17 | GPF_YTICKS     | int   | Ticks on the vertical plot area axis                                | 0-100<br>Not null   |
| 18 | GPF_GRIDSWIDTH | int   | Plot area grids width   | 1-4 or 0 (default)<br>Not null  |
| 19 | GPF_TICKSWIDTH | int   | Plot area ticks width   | 1-4 or 0 (default)<br>Not null  |
| 20 | GPF_GRIDSCOLOR | int   | Color of the grids  | Valid RGB color: <Red,Green,Blue><br>0...2 <sup>24</sup> -1 or -1 (default)<br>Not null   |
| 21 | GPF_TICKSCOLOR | int   | Color of the ticks  | Valid RGB color: <Red,Green,Blue><br>0...2 <sup>24</sup> -1 or -1 (default)<br>Not null   |
| 22 | GPF_BKGNDCOLOR | int   | Color of the plot area background                                   | Valid RGB color: <Red,Green,Blue><br>0...2 <sup>24</sup> -1 or -1 (default)<br>Not null   |
| 23 | GPF_LABELCOLOR | int   | Color of the labels   | Valid RGB color: <Red,Green,Blue><br>0...2 <sup>24</sup> -1 or -1 (default)<br>Not null   |
| 24 | GPF_PARAMETERS | int   | Number of parameters  | 1 ≤ n ≤ 10<br>Modulo(n,2)=0 and two subsequent parameters must have identical GPF_COLOR, GPF_LINEWIDTH, GPF_LINestyle, GPF_LINESYMBOL as they are plotted against each other for GPF_TYPE=3<br>Not null |
| 25 | GPF_STATUS     | ch(1) | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null  |

### 3.1.21. GPFP Table

This table provides the complete list of parameters contained in the graphic displays. It is an extension to the database format delivered by ALTEL.

| No. | Column        | Type   | Description                      | Restriction  |
|-----|---------------|--------|----------------------------------|--|
| 1   | GPFP_NUMBER   | ch(6)  | Name of the GRD                  | Unique, A-Z, a-z or 0-9<br>Not null  |
| 2   | GPFP_PNAME    | ch(6)  | Parameter name                   | Valid parameter name<br>Not null   |
| 3   | GPFP_COMMENT  | vc(32) | Comment related to the parameter | None, not null   |
| 4   | GPFP_POSITION | int    | Position of the parameter        | 0-GPF_PARAMETERS-1 Not null  |
| 5   | GPFP_COMM     | int    | Commutation of the parameter     | >=0: Occurrence number within the telemetry format<br>-1: Every sample (default)<br>Not null |





# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |                 |       |   |   |
|----|-----------------|-------|---|---|
| 6  | GPFP_VALMODE    | ch(1) | Display mode of the parameter value:<br>0 =Default (calibrated)<br>R =Raw (coded)<br>B =Binary (raw, not coded)<br>I =Inverse binary (as above)<br>O=Octal (as above)<br>D =Decimal (as above)<br>H =Hexadecimal (as above) | In set {0,R,B,I,O,D,H}<br>Not null  |
| 7  | GPFP_VALCHARS   | int   | Number of character positions for the parameter value   | >0 or 0 (default)<br>Not null   |
| 8  | GPFP_VALMINIMUM | float | Minimum plot value of the parameter   | Not null  |
| 9  | GPFP_VALMAXIMUM | float | Maximum plot value of the parameter   | Not null  |
| 10 | GPFP_COLOR      | int   | Color of the parameter and its plot line  | Valid RGB color: <Red,Green,Blue><br>0...2 <sup>24</sup> -1 or -1 (default)<br>Not null |
| 11 | GPFP_LINEWIDTH  | int   | Thickness of the plot line  | 1-4 or 0 (default)<br>Not null  |
| 12 | GPFP_LINestyle  | ch(1) | Style of the plot line:<br>0=Solid line<br>1=Dotted line<br>2=Dashed line   | In set {0,1,2}<br>Not null  |
| 13 | GPFP_LINESYMBOL | ch(1) | Symbol on the plot line:<br>0=No symbol<br>1=Point<br>2=Stroke<br>3=Cross<br>4=Square   | In set {0,1,2,3,4}<br>Not null  |

### 3.1.22. PPF Table

This table provides the complete list of parameter observation displays (POD). It is an extension to the database format delivered by ALTEL.

| No. | Column              | Type    | Description   | Restriction  |
|-----|---------------------|---------|---|--|
| 1   | PPF_NUMBER          | ch(6)   | Name of the POD   | Unique, A-Z, a-z or 0-9<br>Not null  |
| 2   | PPF_TITLE           | vc(32)  | Title   | None, not null   |
| 3   | PPF_LAYOUT          | int     | Column layout $n$ :<br>Time=2 <sup>0</sup>   Value=2 <sup>1</sup>   Status=2 <sup>2</sup>   Quality=2 <sup>3</sup>  | 1 <= $n$ < 16 or -1 (default)<br>Not null  |
| 4   | PPF_EXPRESSIONSTACK | vc(256) | Expression stack: $e_1, e_2, e_3, \dots$<br>Constants: $\pi, e$<br>Operators: $/, x, +, -, +/-$<br>Functions: $\sin, \cos, \tan, \arcsin, \arccos, \arctan, \exp, \ln, \log, \sqrt{x^2}, x^y$ | $e_i$ : Parameter name, operator, function, constant or number<br>Parameter format: Tag[i,j]<br>Tag: Parameter name<br>$i$ : Sample (-100...100)<br>$j$ : 0 or 1 (coded)<br>Not null |
| 5   | PPF_EXPRESSIONCOLOR | int     | Color of the expression   | Valid RGB color: <Red,Green,Blue><br>0...2 <sup>24</sup> -1 or -1 (default)<br>Not null  |
| 6   | PPF_PARAMETERS      | vc(256) | Associated parameters: $p_1, p_2, \dots$  | $p_i$ : Valid parameter name<br>Not null   |
| 7   | PPF_PARAMETERSCOLOR | int     | Color of the associated parameters  | Valid RGB color: <Red,Green,Blue><br>0...2 <sup>24</sup> -1 or -1 (default)<br>Not null  |



# BINARY SPACE

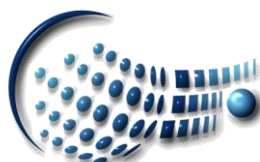
## RELIABLE SPACE SYSTEMS

|    |                    |         |  |   |
|----|--------------------|---------|--|---|
| 8  | PPF_FUNCTIONS      | vc(256) | Applied functions: $f_1, f_2, \dots$<br>Functions: delta, min, max, avg, var, upd, chg | $f_i$ : Function identifier<br>Not null   |
| 9  | PPF_FUNCTIONSCOLOR | int     | Color of the applied functions   | Valid RGB color: <Red,Green,Blue><br>$0 \dots 2^{24} - 1$ or -1 (default)<br>Not null |
| 10 | PPF_STATUS         | ch(1)   | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted                    | In set {N,T,C,D} or null  |

### 3.1.23. CCF Table

This table provides the complete list of telecommand functions.

| No. | Column     | Type    | Description  | Restriction  |
|-----|------------|---------|--|--|
| 1   | CCF_CNAME  | vc(12)  | Name of the telecommand function   | Unique, not null   |
| 2   | CCF_DESCR  | vc(25)  | Description of the function  | Not null   |
| 3   | CCF_CTYPE  | vc(5)   | Telecommand function type:<br>HP =High priority<br>ML =Memory load<br>etc.                                   | In set {INTGT,INTG,HP1,HP2,HPC2,ICS,RBI,HL,ML,ML+,RM,SC,LSC}<br>Not null |
| 4   | CCF_WNDCL  | vc(2)   | Not yet implemented  | Invisible  |
| 5   | CCF_TCPID  | ch(4)   | Associated telecommand packet  | Not null   |
| 6   | CCF_ADD    | ch(2)   | Telecommand identifier   | 00-FF (hex) or null  |
| 7   | CCF_LDESC  | vc(48)  | Detailed description   | None or null   |
| 8   | CCF_DESS   | vc(5)   | Destination sub-system   | None or null   |
| 9   | CCF_DEUN   | vc(5)   | Destination unit   | None or null   |
| 10  | CCF_ROUTS  | vc(5)   | Router sub-system  | None or null   |
| 11  | CCF_ROUTU  | vc(5)   | Router unit  | None or null   |
| 12  | CCF_OPINFO | vc(254) | Operational information  | None or null   |
| 13  | CCF_VALID  | vc(254) | Validity information   | None or null   |
| 14  | CCF_FUNC1  | vc(255) | Description of the data field contents   | None or null   |
| 15  | CCF_FUNC2  | vc(255) | Description of the data field contents (continued)   | None or null   |
| 16  | CCF_ALTC   | vc(128) | Alternative telecommand functions: $f_1, f_2, \dots$   | None or null   |
| 17  | CCF_CMPTC  | vc(128) | Complementary telecommand functions: $f_1, f_2, \dots$   | None or null   |
| 18  | CCF_SSCH   | vc(4)   | Sub-schedule destination   | None or null   |
| 19  | CCF_TCTYPE | ch(1)   | Telecommand type:<br>I =Immediate<br>T =Time-tagged<br>S=First step (of 2-step)<br>2=Second step (of 2-step) | In set {I,T,S,2} or null   |
| 20  | CCF_ULPPTF | int     | Not yet implemented  | Invisible  |
| 21  | CCF_HAZ    | ch(1)   | Flag indicating hazardous operation:<br>V =Vital<br>Y =Potentially hazardous<br>N=Non-critical               | In set {V,Y,N}<br>Not null   |
| 22  | CCF_COSEV  | vc(10)  | Mission planner event  | None or null   |



|    |             |        |   |                          |
|----|-------------|--------|---|--------------------------|
| 23 | CCF_STATUS  | ch(1)  | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null |
| 24 | CCF_SC_NAME | vc(32) | Spacecraft name   | None or null             |
| 25 | CCF_CVPID   | vc(12) | Execution verification parameter group                              | Valid group name or null |
| 26 | CCF_PVPID   | vc(12) | Pre-execution verification parameter group                          | Valid group name or null |

### 3.1.24. CDF Table

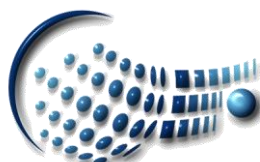
This table provides the complete list of parameters contained in the telecommand functions.

| No. | Column     | Type   | Description   | Restriction                                  |
|-----|------------|--------|---|--|
| 1   | CDF_CNAME  | vc(12) | Name of the telecommand function  | Valid telecommand function<br>Not null       |
| 2   | CDF_ELTYPE | ch(1)  | Element type:<br>P=Variable parameter<br>F=Fixed value                    | In set {P,F}<br>Not null                     |
| 3   | CDF_ELLEN  | int    | Element bit length:<br>Size of parameter or length of the fixed value     | 1-48<br>Not null                             |
| 4   | CDF_BITOFF | int    | Element bit offset  | 0-1919<br>Not null                           |
| 5   | CDF_PNAME  | vc(9)  | Telecommand parameter associated with this element                        | Valid telecommand parameter or null          |
| 6   | CDF_VTYPE  | ch(1)  | Parameter value type:<br>C=Calibrated value<br>U=Raw value                | In set {C,U}<br>Not null                     |
| 7   | CDF_VALUE  | vc(32) | Element value:<br>Value of parameter or constant                          | Not null if CDF_ELTYPE=F or else may be null |
| 8   | CDF_RADIX  | ch(1)  | Element value radix:<br>B=Binary<br>O=Octal<br>D=Decimal<br>H=Hexadecimal | In set {B,O,D,H} or null                     |
| 9   | CDF_DESCR  | vc(48) | Description of the element  | None or null                                 |
| 10  | CDF_STATUS | ch(1)  | Record change status:<br>N=New<br>C=Changed<br>D=Deleted                  | In set {N,C,D} or null                       |

### 3.1.25. TCP Table

This table provides the complete list of telecommand packets.

| No. | Column    | Type   | Description                              | Restriction              |
|-----|-----------|--------|--|--------------------------|
| 1   | TCP_ID    | ch(4)  | Telecommand packet name                  | Unique, not null         |
| 2   | TCP_TYPE  | ch(1)  | Telecommand packet type                  | Must be 1, not null      |
| 3   | TCP_DESCR | vc(32) | Description of the telecommand packet    | None, not null           |
| 4   | TCP_HFLG  | int    | Data field header flag:<br>0=No<br>1=Yes | In set {0,1}<br>Not null |



# BINARY SPACE

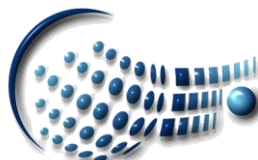
## RELIABLE SPACE SYSTEMS

|    |              |         |  |  |
|----|--------------|---------|--|--|
| 5  | TCP_APID     | ch(3)   | Application identifier   | In set {001,002,003,004,011,012,013,020,021,022,7FE}<br>Not null |
| 6  | TCP LENG     | int     | Packet data field size (in bytes) minus 1                            | 0-241<br>Not null  |
| 7  | TCP_FUNC     | vc(254) | Extended description   | None or null   |
| 8  | TCP_FUNC2    | vc(254) | Extended description (continued)                                     | None or null   |
| 9  | TCP_STATUS   | ch(1)   | Record change status:<br>N=New<br>T =Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null   |
| 10 | TCP_SC_NAME  | vc(32)  | Spacecraft name  | None or null   |
| 11 | TCP_UNIQID   | int     | Not used   | Invisible  |
| 12 | TCP_VARIABLE | int     | Flag indicating variable length:<br>0=Fixed<br>1=Variable            | In set {0,1}<br>Not null   |

### 3.1.26. PCDF Table

This table provides the complete list of parameters related to the telecommand packets.

| No. | Column      | Type   | Description  | Restriction   |
|-----|-------------|--------|--|---|
| 1   | PCDF_CNAME  | ch(4)  | Name of the telecommand packet   | Valid telecommand packet<br>Not null                    |
| 2   | PCDF_ELTYPE | ch(1)  | Element type:<br>P=Variable parameter<br>F=Fixed value   | In set {P,F}<br>Not null                                |
| 3   | PCDF_ELLEN  | int    | Element bit length:<br>Size of parameter or length of the fixed value  | 1-48<br>Not null  |
| 4   | PCDF_BITOFF | int    | Element bit offset   | 0-1919<br>Not null                                      |
| 5   | PCDF_PNAME  | vc(9)  | Telecommand parameter associated with this element   | Valid telecommand parameter or null                     |
| 6   | PCDF_VTYPE  | ch(1)  | Parameter value type:<br>C=Calibrated value<br>U=Raw value   | In set {C,U}<br>Not null                                |
| 7   | PCDF_VALUE  | vc(17) | Element value:<br>Value of parameter or constant   | Not null if CDF_ELTYPE=F or else may be null            |
| 8   | PCDF_RADIX  | ch(1)  | Element value radix:<br>B =Binary<br>O=Octal<br>D =Decimal<br>H =Hexadecimal   | In set {B,O,D,H} or null                                |
| 9   | PCDF_DESCR  | vc(48) | Description of the element   | None or null  |
| 10  | PCDF_STATUS | ch(1)  | Record change status:<br>N=New<br>C=Changed<br>D=Deleted   | In set {N,C,D} or null                                  |
| 11  | PCDF_CODING | ch(1)  | Type of digital coding:<br>B =Bit pattern<br>I =Signed integer<br>U=Unsigned integer<br>S =Complement complex<br>1 =Floating point | In set {B,I,U,S,1} if PCDF_ELTYPE=P or else may be null |





### 3.1.27. CPC Table

This table provides the complete list of telecommand parameters.

| No. | Column      | Type   | Description  | Restriction  |
|-----|-------------|--------|--|--|
| 1   | CPC_PNAME   | vc(9)  | Telecommand parameter name   | Unique, not null   |
| 2   | CPC_DESCR   | vc(48) | Description of the telecommand parameter   | None, not null   |
| 3   | CPC_LENGTH  | vc(2)  | Parameter width (in bits)  | 1-48<br>Not null   |
| 4   | CPC_DSPFMT  | int    | Not yet implemented  | Invisible  |
| 5   | CPC_CODING  | ch(1)  | Type of digital coding:<br>B = Bit pattern<br>I = Signed integer<br>U = Unsigned integer<br>S = Complement complex<br>1 = Floating point | In set {B,I,U,S,1}<br>Not null                               |
| 6   | CPC_UNIT    | vc(4)  | Parameter unit   | None or null   |
| 7   | CPC_CATEG   | ch(1)  | Parameter calibration:<br>N = Numerical<br>S = Status<br>R = None  | In set {N,S,R}<br>Not null                                   |
| 8   | CPC_CALREF  | vc(8)  | Attached numerical calibration curve   | Valid numerical calibration curve if<br>CPC_CATEG=N or null  |
| 9   | CPC_TXFREF  | vc(8)  | Attached textual calibration curve   | Valid textual calibration curve if<br>CPC_CATEG=S or null    |
| 10  | CPC_OCFREF  | vc(10) | Attached numerical out-of-limit table  | Valid numerical out-of-limit table if<br>CPC_CATEG=N or null |
| 11  | CPC_FXFREF  | vc(10) | Attached textual out-of-limit table  | Valid textual out-of-limit table if<br>CPC_CATEG=S or null   |
| 12  | CPC_STATUS  | ch(1)  | Record change status:<br>N = New<br>T = Draft<br>C = Changed<br>D = Deleted  | In set {N,T,C,D} or null                                     |
| 13  | CPC_SC_NAME | vc(32) | Spacecraft name  | None or null   |

### 3.1.28. PCPS Table

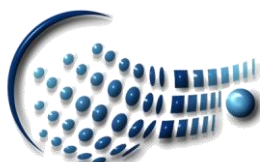
This table provides the complete list of all associations between the parameter sets and telecommand functions.

| No. | Column     | Type   | Description                                      | Restriction                                 |
|-----|------------|--------|--|---|
| 1   | PCPS_CNAME | vc(12) | Name of the telecommand function                 | Valid telecommand function<br>Not null      |
| 2   | PCPS_NAME  | vc(9)  | Name of the associated telecommand parameter set | Valid telecommand parameter set<br>Not null |

### 3.1.29. CPS Table

This table provides the complete list of telecommand parameter sets.

| No. | Column    | Type   | Description                                  | Restriction      |
|-----|-----------|--------|--|------------------|
| 1   | CPS_NAME  | vc(9)  | Telecommand parameter set name               | Unique, not null |
| 2   | CPS_DESCR | vc(32) | Description of the telecommand parameter set | None or null     |



|   |            |       |  |                          |
|---|------------|-------|--|--------------------------|
| 3 | CPS_STATUS | ch(1) | Record change status:<br>N=New<br>T =Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null |
|---|------------|-------|--|--------------------------|

### 3.1.30. CPSP Table

This table provides the complete list of parameters contained in the telecommand parameter sets.

| No. | Column      | Type   | Description  | Restriction                                 |
|-----|-------------|--------|--|---|
| 1   | CPSP_NAME   | vc(9)  | Name of the telecommand parameter set  | Valid telecommand parameter set<br>Not null |
| 2   | CPSP_PNAME  | vc(9)  | Name of the associated telecommand parameter                                   | Valid telecommand parameter<br>Not null     |
| 3   | CPSP_BITOFF | int    | Parameter bit offset   | 0-1919<br>Not null                          |
| 4   | CPSP_PVALUE | vc(32) | Parameter value  | None or null                                |
| 5   | CPSP_VTYPE  | ch(1)  | Parameter value type:<br>C=Calibrated value<br>U=Raw value                     | In set {C,U}<br>Not null                    |
| 6   | CPSP_RADIX  | ch(1)  | Parameter value radix:<br>B =Binary<br>O=Octal<br>D =Decimal<br>H =Hexadecimal | In set {B,O,D,H} or null                    |

### 3.1.31. PVP Table

This table provides the complete list of pre-execution verification parameter groups.

| No. | Column     | Type    | Description   | Restriction              |
|-----|------------|---------|---|--------------------------|
| 1   | PVP_ID     | vc(12)  | Telecommand pre-execution verification group name                         | Unique, not null         |
| 2   | PVP_DESCR  | vc(255) | Description of the telecommand pre-execution verification parameter group | None or null             |
| 3   | PVP_STATUS | ch(1)   | Record change status:<br>N=New<br>T =Draft<br>C=Changed<br>D=Deleted      | In set {N,T,C,D} or null |

### 3.1.32. PEV Table

This table provides the complete list of parameters contained in the pre-execution verification parameter groups.

| No. | Column     | Type   | Description  | Restriction  |
|-----|------------|--------|--|--|
| 1   | PEV_ID     | vc(12) | Name of the telecommand pre-execution verification group   | Valid telecommand pre-execution verification group<br>Not null |
| 2   | PEV_PNAME  | ch(6)  | Telemetry parameter used for the verification              | Valid telemetry parameter<br>Not null                          |
| 3   | PEV_PVALUE | vc(17) | Parameter value  | None, not null   |
| 4   | PEV_INTER  | ch(1)  | Parameter value type:<br>C=Calibrated value<br>U=Raw value | In set {C,U}<br>Not null                                       |



### 3.1.33. CVP Table

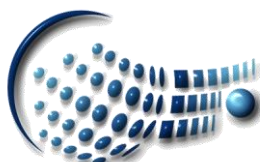
This table provides the complete list of execution verification parameter groups.

| No. | Column     | Type    | Description   | Restriction   |
|-----|------------|---------|---|---|
| 1   | CVP_ID     | vc(12)  | Telecommand execution verification group name                       | Unique, not null  |
| 2   | CVP_NAME   | vc(32)  | Operational telecommand execution verification group name           | None, not null  |
| 3   | CVP_DESCR  | vc(255) | Description of the telecommand execution verification group         | None or null  |
| 4   | CVP_MINWND | ch(1)   | Not yet implemented   | Invisible   |
| 5   | CVP_NMBST  | int     | Number of stages  | Must correspond with the maximum of CEV_STAGE<br>Not null |
| 6   | CVP_STATUS | ch(1)   | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null                                  |

### 3.1.34. CEV Table

This table provides the complete list of parameters contained in the execution verification parameter groups.

| No. | Column      | Type   | Description   | Restriction   |
|-----|-------------|--------|---|---|
| 1   | CEV_ID      | vc(12) | Name of the telecommand execution verification group                          | Valid telecommand execution verification group<br>Not null                                      |
| 2   | CEV_EVTYPE  | ch(1)  | Execution verification mode:<br>1=Exception<br>3=Report<br>4=Dump<br>S=Status | In set {1,3,4,S}<br>Not null  |
| 3   | CEV_PNAME   | ch(6)  | Telemetry parameter used for the verification                                 | Valid telemetry parameter<br>Not null   |
| 4   | CEV_PVALUE  | vc(17) | Parameter value   | Valid value or telecommand parameter name if CEV_INTER=E (status consistency check)<br>Not null |
| 5   | CEV_INTER   | ch(1)  | Parameter value type:<br>C=Calibrated value<br>U=Raw value<br>E=Echoed        | In set {C,U,E}<br>Not null  |
| 6   | CEV_SWND    | int    | Time window for the verification stage (in seconds)                           | >=0<br>Not null   |
| 7   | CEV_STAGE   | vc(6)  | Verification stage  | >0<br>Not null  |
| 8   | CEV_EPACKID | ch(4)  | Event packet name   | None or null  |
| 9   | CEV_STATUS  | ch(1)  | Record change status:<br>N=New<br>C=Changed<br>D=Deleted                      | In set {N,C,D} or null  |



### 3.1.35. CSF Table

This table provides the complete list of telecommand sequences.

| No. | Column        | Type   | Description  | Restriction              |
|-----|---------------|--------|--|--------------------------|
| 1   | CSF_NAME      | vc(16) | Telecommand sequence name  | Unique, not null         |
| 2   | CSF_DESCR     | vc(64) | Description of the telecommand sequence                              | None or null             |
| 3   | CSF_IFTT      | ch(1)  | Not yet implemented  | Invisible                |
| 4   | CSF_NFPARS    | int    | Number of parameters for this sequence                               | >=0<br>Not null          |
| 5   | CSF_NSTMNT    | int    | Number of entries for this sequence                                  | >0<br>Not null           |
| 6   | CSF_REFCNT    | int    | Not yet implemented  | Invisible                |
| 7   | CSF_TYPCNT    | int    | Not yet implemented  | Invisible                |
| 8   | CSF_TYPE      | ch(1)  | Not yet implemented  | Invisible                |
| 9   | CSF_SEQ_DATE  | vc(32) | Date and time when the sequence was created                          | None or null             |
| 10  | CSF_SEQ_USER  | vc(32) | Name of the person that created the sequence                         | None or null             |
| 11  | CSF_DOC_NAME  | vc(64) | Name of the document the sequence was created from                   | None or null             |
| 12  | CSF_DOC_ISSUE | vc(10) | Issue of that document   | None or null             |
| 13  | CSF_DOC_DATE  | vc(32) | Issue date of that document  | None or null             |
| 14  | CSF_COSEV     | vc(10) | Mission planner event  | None or null             |
| 15  | CSF_SSCH      | vc(4)  | Sub-schedule destination   | None or null             |
| 16  | CSF_STATUS    | ch(1)  | Record change status:<br>N=New<br>T =Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null |

### 3.1.36. CSS Table

This table provides the complete list of functions, sequences or procedures contained in the telecommand sequences.

| No. | Column      | Type   | Description   | Restriction  |
|-----|-------------|--------|---|--|
| 1   | CSS_SQNAME  | vc(16) | Name of the telecommand sequence  | Valid telecommand sequence<br>Not null   |
| 2   | CSS_COMMENT | vc(64) | Description of the telecommand sequence entry   | None or null   |
| 3   | CSS_TCTYPE  | vc(1)  | Sequence entry type:<br>I = Immediate<br>T = Time-tagged<br>S = First step (of 2-step)<br>2 = Second step (of 2-step) | In set {I,T,S,2}<br>Not null   |
| 4   | CSS_1STEP   | vc(12) | Associated first step   | Valid telecommand sequence entry with<br>CSS_TCTYPE=S when CSS_TCTYPE=2 else<br>null |
| 5   | CSS_EXTRA1  | vc(40) | Additional information (part 1)   | None or null   |
| 6   | CSS_EXTRA2  | vc(40) | Additional information (part 2)   | None or null   |
| 7   | CSS_EXTRA3  | vc(40) | Additional information (part 3)   | None or null   |
| 8   | CSS_EXTRA4  | vc(40) | Additional information (part 4)   | None or null   |
| 9   | CSS_EXTRA5  | vc(40) | Additional information (part 5)   | None or null   |
| 10  | CSS_ENTRY   | int    | Sequence entry  | >0<br>Not null   |





# BINARY SPACE

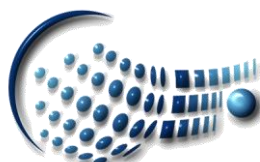
## RELIABLE SPACE SYSTEMS

|    |             |        |   |   |
|----|-------------|--------|---|---|
| 11 | CSS_IDENT   | vc(12) | Name of the entry in the sequence   | Valid telecommand function, sequence or procedure |
| 12 | CSS_TYPE    | vc(1)  | Sequence entry type:<br>C=Telecommand function<br>S=Telecommand sequence<br>P=Telecommand procedure | In set {C,S,P}<br>Not null                        |
| 13 | CSS_RELTIME | vc(21) | Release time for this entry<br>Delta time from the start of the sequence<br>Format: +HH:MM:SS       | None or null                                      |
| 14 | CSS_ABSTIME | vc(24) | Execution time for this entry<br>Absolute time<br>Format: DD/MM/YYYY HH:MM:SS                       | Valid (future) time if CSS_TCTYPE=T else null     |
| 15 | CSS_EXETIME | vc(21) | Execution time for this entry<br>Delta time from the start of the sequence<br>Format: +HH:MM:SS     | None or null                                      |
| 16 | CSS_LABEL   | vc(5)  | Not yet implemented   | Invisible   |
| 17 | CSS_GROUP   | vc(8)  | Grouping flag:<br>G for non-first elements in the group   | In set {G} or null                                |
| 18 | CSS_BLOCK   | vc(3)  | Blocking flag:<br>B <sub>n</sub> for non-first elements in the block                                | B <sub>n</sub> , n>0 or null                      |
| 19 | CSS_BLKCNTL | ch(1)  | Blocking flag:<br>B for first block element   | In set {B} or null                                |
| 20 | CSS_ILOCK   | ch(1)  | Not yet implemented   | Invisible   |
| 21 | CSS_PTV     | ch(1)  | Not yet implemented   | Invisible   |
| 22 | CSS_PTV_V   | ch(1)  | Not yet implemented   | Invisible   |
| 23 | CSS_PTV_C   | ch(1)  | Not yet implemented   | Invisible   |
| 24 | CSS_PTV_F   | ch(1)  | Not yet implemented   | Invisible   |
| 25 | CSS_UV      | ch(1)  | Not yet implemented   | Invisible   |
| 26 | CSS_NPARS   | int    | Number of parameters  | >=0<br>Not null                                   |
| 27 | CSS_CEV     | ch(1)  | Not yet implemented   | Invisible   |

### 3.1.37. CSP Table

This table provides the complete list of formal parameters related to the telecommand sequences.

| No. | Column     | Type   | Description  | Restriction                            |
|-----|------------|--------|--|--|
| 1   | CSP_SQNAME | vc(16) | Name of the telecommand sequence                     | Valid telecommand sequence<br>Not null |
| 2   | CSP_FPNAME | vc(12) | Formal name for the sequence parameter               | Not null                               |
| 3   | CSP_FPNUM  | int    | Position number of the parameter within the sequence | >=0<br>Not null                        |
| 4   | CSP_DESCR  | vc(64) | Parameter description                                | None or null                           |
| 5   | CSP_LENGTH | int    | Parameter width (in bits)                            | >0<br>Not null                         |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |            |        |  |   |
|----|------------|--------|--|---|
| 6  | CSP_CODING | vc(1)  | Type of digital coding:<br>B = Bit pattern<br>I = Signed integer<br>U = Unsigned integer<br>S = Complement complex<br>1 = Floating point | In set {B,I,U,S,1}<br>Not null                            |
| 7  | CSP_UNIT   | vc(4)  | Parameter unit   | None or null  |
| 8  | CSP_VTYPE  | vc(1)  | Parameter value type:<br>V = Custom (interactive)<br>A = Calibrated value  | In set {V,A}<br>Not null                                  |
| 9  | CSP_RADIX  | vc(1)  | Parameter value radix:<br>B = Binary<br>O = Octal<br>D = Decimal<br>H = Hexadecimal  | In set {B,O,D,H}<br>Not null                              |
| 10 | CSP_DEFVAL | vc(32) | Default parameter value  | None or null  |
| 11 | CSP_CATEG  | vc(1)  | Parameter calibration:<br>N = Numerical<br>S = Status<br>R = None  | In set {N,S,R}<br>Not null                                |
| 12 | CSP_CALREF | vc(8)  | Attached numerical calibration curve   | Valid numerical calibration curve if CSP_CATEG=N or null  |
| 13 | CSP_TXFREF | vc(8)  | Attached textual calibration curve   | Valid textual calibration curve if CSP_CATEG=S or null    |
| 14 | CSP_OCFREF | vc(10) | Attached numerical out-of-limit table  | Valid numerical out-of-limit table if CSP_CATEG=N or null |
| 15 | CSP_FXFREF | vc(10) | Attached textual out-of-limit table  | Valid textual out-of-limit table if CSP_CATEG=S or null   |

### 3.1.38. SDF Table

This table provides the complete list of parameters contained in the telecommand sequence entries.

| No. | Column       | Type   | Description  | Restriction                             |
|-----|--------------|--------|--|---|
| 1   | SDF_SQNAME   | vc(16) | Name of the telecommand sequence   | Valid telecommand sequence<br>Not null  |
| 2   | SDF_ENTRY    | int    | Sequence entry to which the parameter belongs to                                       | >0<br>Not null                          |
| 3   | SDF_BITOFF   | int    | Parameter bit offset   | 0-1919<br>Not null                      |
| 4   | SDF_PNAME    | vc(9)  | Parameter name   | Valid telecommand parameter<br>Not null |
| 5   | SDF_VALRADIX | vc(1)  | Parameter value radix:<br>B = Binary<br>O = Octal<br>D = Decimal<br>H = Hexadecimal    | In set {B,O,D,H}<br>Not null            |
| 6   | SDF_DSPFMT   | vc(2)  | Not yet implemented  | Invisible                               |
| 7   | SDF_VALTYPE  | vc(1)  | Parameter value type:<br>F = Formal parameter<br>A = Calibrated value<br>V = Raw value | In set {F,A,V} or null                  |
| 8   | SDF_VALUE    | vc(32) | Default parameter value  | None, not null                          |
| 9   | SDF_COMMENT  | vc(40) | Parameter comment  | None or null                            |



|    |            |        |                                       |                                   |
|----|------------|--------|---------------------------------------|-----------------------------------|
| 10 | SDF_PSET   | vc(9)  | Name of the associated memory area    | Valid memory area or null         |
| 11 | SDF_PSINST | vc(10) | Version of the associated memory area | Valid memory area version or null |

### 3.1.39. BCF Table

This table provides the complete list of telecommand blocks.

| No. | Column      | Type   | Description   | Restriction                                |
|-----|-------------|--------|---|--|
| 1   | BCF_CNAME   | vc(12) | Telecommand block name  | Unique, not null                           |
| 2   | BCF_INTER   | ch(3)  | Block type  | In set {LI,LTT,L2S,NI,NTT,N2S}<br>Not null |
| 3   | BCF_MAXLEN  | int    | Maximum block length (in bytes)                                     | 0-240<br>Not null                          |
| 4   | BCF_BYTEOFF | int    | Byte offset in the data field                                       | 1-240<br>Not null                          |
| 5   | BCF_STATUS  | ch(1)  | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null                   |

### 3.1.40. BCCF Table

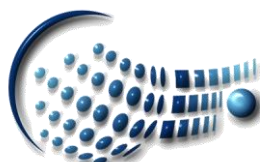
This table provides the complete list of functions contained in the telecommand blocks.

| No. | Column       | Type   | Description   | Restriction  |
|-----|--------------|--------|---|--|
| 1   | BCCF_CNAME   | vc(12) | Name of the telecommand block   | Valid telecommand block<br>Not null  |
| 2   | BCCF_INTER   | ch(1)  | Block entry type:<br>T=Telecommand function<br>P=Alternative telecommand packet<br>C=Alternative telecommand type | In set {T,P,C}<br>Not null   |
| 3   | BCCF_CONTENT | vc(12) | Block entry command   | Valid telecommand function, alternative telecommand packet or alternative telecommand type<br>Not null |

### 3.1.41. GEN Table

This table provides the complete list of on-board processors.

| No. | Column        | Type   | Description                                      | Restriction                            |
|-----|---------------|--------|--|--|
| 1   | GEN_PROCESSOR | vc(10) | On-board processor name                          | Unique, not null                       |
| 2   | GEN_PATCHTC   | vc(12) | Telecommand function used to patch the processor | Valid telecommand function or null     |
| 3   | GEN_DUMPPTC   | vc(12) | Telecommand function used to dump the processor  | Valid telecommand function<br>Not null |
| 4   | GEN_TMPKT     | vc(4)  | Telemetry packet in which the dump arrives       | Valid telemetry packet<br>Not null     |
| 5   | GEN_STARTPAGE | int    | Page number where the processor memory starts    | 0-15<br>Not null                       |
| 6   | GEN_STARTADDR | int    | Address where the processor memory starts        | 0-65535<br>Not null                    |
| 7   | GEN_ENDPAGE   | int    | Page number where the processor memory ends      | 0-15<br>Not null                       |



|    |              |       |   |                                |
|----|--------------|-------|---|--------------------------------|
| 8  | GEN_ENDADDR  | int   | Address where the processor memory ends                             | 0-65535<br>Not null            |
| 9  | GEN_IMGNAME  | vc(9) | Name of the master image for this processor                         | Valid memory image<br>Not null |
| 10 | GEN_INTERVAL | int   | Interval in seconds between dump packets                            | >0<br>Not null                 |
| 11 | GEN_NUMPATCH | int   | Number of patch commands between each dump command                  | >=0 or null                    |
| 12 | GEN_STATUS   | ch(1) | Record change status:<br>N=New<br>T=Draft<br>C=Changed<br>D=Deleted | In set {N,T,C,D} or null       |

### 3.1.42. MAD Table

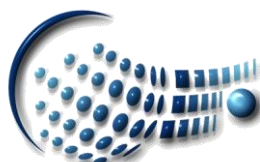
This table provides the complete list of all memory areas contained in the on-board processors.

| No. | Column        | Type   | Description  | Restriction                          |
|-----|---------------|--------|--|--------------------------------------|
| 1   | MAD_PROCESSOR | vc(10) | Name of the on-board processor                                 | Valid on-board processor<br>Not null |
| 2   | MAD_NAME      | vc(9)  | Memory area name   | Unique, not null                     |
| 3   | MAD_DESCR     | vc(64) | Description of the memory area                                 | None or null                         |
| 4   | MAD_TYPE      | vc(4)  | Memory area type:<br>MEM=Memory block<br>IMG=Memory image      | In set {MEM,IMG}<br>Not null         |
| 5   | MAD_PATCHTC   | vc(12) | Telecommand function used to patch the memory area             | Valid telecommand function or null   |
| 6   | MAD_DUMPPTC   | vc(12) | Telecommand function used to dump the memory area              | Valid telecommand function or null   |
| 7   | MAD_TMPKT     | vc(4)  | Telemetry packet in which the dump arrives                     | Valid telemetry packet or null       |
| 8   | MAD_PATCHABLE | ch(1)  | Flag indicating if the memory area can be patched              | In set {Y,N}<br>Not null             |
| 9   | MAD_DUMPABLE  | ch(1)  | Flag indicating if the memory area can be dumped               | In set {Y,N}<br>Not null             |
| 10  | MAD_UPDATE    | ch(1)  | Flag indicating if the ground image can be dynamically updated | In set {Y,N}<br>Not null             |
| 11  | MAD_INHIBIT   | ch(1)  | Flag indicating if the dynamic update can be inhibited         | In set {Y,N}<br>Not null             |
| 12  | MAD_STATUS    | ch(1)  | Record change status:<br>N=New<br>C=Changed<br>D=Deleted       | In set {N,C,D} or null               |

### 3.1.43. MEM Table

This table provides the complete list of all memory blocks contained in the memory areas.

| No. | Column     | Type  | Description  | Restriction                         |
|-----|------------|-------|--|-------------------------------------|
| 1   | MEM_MNAME  | vc(9) | Name of the memory area  | Valid memory area<br>Not null       |
| 2   | MEM_TMNAME | vc(6) | Telemetry parameter that represents the memory block           | Valid telemetry parameter or null   |
| 3   | MEM_TCNAME | vc(9) | Telecommand parameter that is associated with the memory block | Valid telecommand parameter or null |





|   |             |        |                                      |                     |
|---|-------------|--------|--------------------------------------|---------------------|
| 4 | MEM_PAGE    | int    | Memory page this block resides in    | 0-15<br>Not null    |
| 5 | MEM_ADDRESS | int    | Memory address this block resides in | 0-65535<br>Not null |
| 6 | MEM_BITOFF  | int    | Bit offset of the data               | 0-15<br>Not null    |
| 7 | MEM_WIDTH   | int    | Width (in bits) of the data          | 1-32 or null        |
| 8 | MEM_DESCR   | vc(72) | Description of the memory area block | None or null        |

### 3.1.44. IMG Table

This table provides the complete list of all memory images contained in the memory areas.

| No. | Column      | Type   | Description  | Restriction                              |
|-----|-------------|--------|--|--|
| 1   | IMG_MNAME   | vc(9)  | Name of the memory image                                 | None, not null                           |
| 2   | IMG_VERSION | vc(10) | Version of the memory image                              | None<br>Not null                         |
| 3   | IMG_PAGE    | int    | Memory page this image resides in                        | 0-15<br>Not null                         |
| 4   | IMG_ADDRESS | int    | Memory address this image resides in                     | 0-65535<br>Not null                      |
| 5   | IMG_DATA    | text   | Data   | Maximal 2000 hexadecimal numbers or null |
| 6   | IMG_STATUS  | ch(1)  | Record change status:<br>N=New<br>C=Changed<br>D=Deleted | In set {N,C,D} or null                   |

### 3.1.45. CRC Table

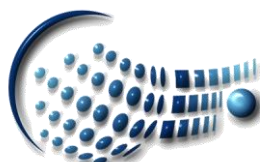
This table provides the complete list of all check sums attached to the memory areas.

| No. | Column        | Type   | Description  | Restriction                          |
|-----|---------------|--------|--|--------------------------------------|
| 1   | CRC_NAME      | vc(12) | Name of the memory area this check sum applies to                    | Valid memory area<br>Not null        |
| 2   | CRC_PROCESSOR | vc(10) | Name of the on-board processor that contains the related memory area | Valid on-board processor<br>Not null |
| 3   | CRC_DEST      | int    | The destination of the check sum                                     | 0-65535 or null                      |
| 4   | CRC_LENGTH    | int    | Length of the check sum  | 16, 32 or null                       |
| 5   | CRC_PAGE      | int    | Memory page the check sum applies to                                 | 0-15 or null                         |
| 6   | CRC_START     | int    | Starting memory address the check sum applies to                     | 0-65535 or null                      |
| 7   | CRC_END       | int    | Ending memory address the check sum applies to                       | 0-65535 or null                      |
| 8   | CRC_SEED      | vc(8)  | Seed value of the check sum  | Value (hex) or null                  |

### 3.1.46. CAT Table

This table provides the complete list of all catalogued memory area versions.

| No. | Column    | Type   | Description                        | Restriction                   |
|-----|-----------|--------|------------------------------------|-------------------------------|
| 1   | CAT_ID    | vc(12) | Catalogue name                     | None, not null                |
| 2   | CAT_NAME  | vc(9)  | Name of the memory area            | Valid memory area<br>Not null |
| 3   | CAT_DESCR | vc(32) | Description of the catalogue entry | None or null                  |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

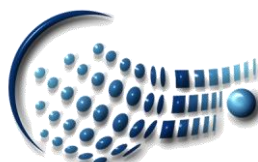
|   |              |        |  |                      |
|---|--------------|--------|--|----------------------|
| 4 | CAT_VERSION  | vc(10) | Version of the memory area catalogued              | None<br>Not null     |
| 5 | CAT_DATE     | vc(32) | Date the memory area was catalogued                | None or null         |
| 6 | CAT_TIME     | vc(32) | Time the memory area was catalogued                | None or null         |
| 7 | CAT_LOCATION | vc(12) | Storage location of the memory area                | None or null         |
| 8 | CAT_COMPLETE | ch(1)  | Flag indicating if the catalogue entry is complete | In set {Y,N} or null |



### 3.2. SCOS-2000 Standard

The following chapters describe the tables and business rules used by a SCOS-2000 based database:

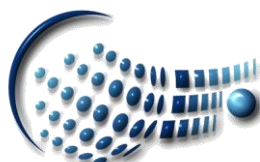
| Chapter | Table | Description  | Page |
|---------|-------|--|------|
| 3.2.1.  | VDF   | Database version definition file, containing the name, the description of the database version including release and issue and the domain identifier         | 34   |
| 3.2.2.  | PCF   | Parameter characteristics file, containing the definition of monitoring parameters   | 34   |
| 3.2.3.  | CUR   | Calibration curve table definitions  | 35   |
| 3.2.4.  | CAF   | Calibration curve file, defining the numerical calibration curves  | 36   |
| 3.2.5.  | CAP   | Calibration curve definition file, defining all the raw/engineering value couples for each numerical calibration curves                                      | 36   |
| 3.2.6.  | TXF   | Text strings calibration curve file, defining the textual calibration curves   | 36   |
| 3.2.7.  | TXP   | Text strings calibration curve definition file, defining all the raw/string value couples for each textual calibration curves                                | 37   |
| 3.2.8.  | MCF   | Polynomial calibration curve definitions, defining the coefficients of the polynomial function used for calibration  | 37   |
| 3.2.9.  | LGF   | Logarithmic calibration curve definitions, defining the coefficients of the logarithmic function used for calibration  | 37   |
| 3.2.10. | OCF   | Out-of-limits checks file, defining the characteristics of all the checks applied to a specified monitoring parameter  | 38   |
| 3.2.11. | OCP   | Out-of-limits definition file, defining the allowed (ranges of) values for monitoring parameters   | 38   |
| 3.2.12. | PID   | Packet identification file, containing the definition of telemetry packets and their correspondence with the packet identification fields                    | 39   |
| 3.2.13. | PIC   | Packet identification criteria file, containing the definition and position of the additional identification fields for each packet type/subtype combination | 40   |
| 3.2.14. | TPCF  | Telemetry packets characteristics file, defining the attributes of those packets   | 41   |
| 3.2.15. | PLF   | Parameter location file, defining the location of the parameters in the fixed size telemetry packets   | 41   |
| 3.2.16. | VPD   | Variable packet definition file, detailing the contents of variable size telemetry packets   | 41   |
| 3.2.17. | GRP   | Parameters and packets groups characteristics file, containing the definition of monitoring parameters and telemetry packets groups                          | 43   |
| 3.2.18. | GRPA  | Parameters groups file, defining the groups of monitoring parameters   | 43   |
| 3.2.19. | GRPK  | Packets groups file, defining the groups of telemetry packets  | 43   |
| 3.2.20. | DPF   | Alphanumeric display proforma file, containing the list of defined displays (ANDs)   | 44   |
| 3.2.21. | DPC   | Alphanumeric display proforma definition file, containing the list of monitoring parameters to be displayed in each AND                                      | 44   |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|         |      |  |    |
|---------|------|--|----|
| 3.2.22. | GPF  | Graphic display proforma file, containing the list of defined displays (GRDs)  | 46 |
| 3.2.23. | GPC  | Graphic display proforma definition file, containing the list of monitoring parameters to be displayed in each GRD                     | 47 |
| 3.2.24. | SPF  | Parameter scrolling display proforma file, containing the list of defined displays (PODs)  | 49 |
| 3.2.25. | SPC  | Parameter scrolling display proforma definition file, containing the list of monitoring parameters to be displayed in each POD         | 49 |
| 3.2.26. | TCP  | Packet header file which defines telecommand packet headers  | 51 |
| 3.2.27. | PCPC | Packet header parameter characteristics file which defines the telecommand packet header parameters                                    | 51 |
| 3.2.28. | PCDF | Packet headers definition file which defines the structure of each telecommand packet header   | 51 |
| 3.2.29. | CCF  | Command characteristics file which defines the telecommands  | 52 |
| 3.2.30. | DST  | Command routing table which defines the destination of the telecommands  | 53 |
| 3.2.31. | CPC  | Command parameter characteristics which defines the editable telecommand parameters  | 53 |
| 3.2.32. | CDF  | Command details file which defines the structure of the telecommand application data field   | 55 |
| 3.2.33. | CSF  | Command sequence file which defines the telecommand sequences  | 55 |
| 3.2.34. | CSS  | Command sequence set which defines the elements (telecommands or telecommand sequences) used in a telecommand sequence                 | 56 |
| 3.2.35. | SDF  | Command sequence details file which defines the values for the editable parameters of all elements contained in a telecommand sequence | 59 |
| 3.2.36. | CSP  | Command sequence parameter file which defines the telecommand sequence (formal) parameter  | 61 |
| 3.2.37. | PTV  | Command pre-transmission validation file which defines the monitoring parameter and value pairs to satisfy validation                  | 62 |
| 3.2.38. | CVS  | Command verification stages file which defines the verification stage details  | 63 |
| 3.2.39. | CVE  | Command verification expression file which defines the monitoring parameter and value pairs to satisfy verification                    | 63 |
| 3.2.40. | CVP  | Command verification profiles file which defines the mapping of verification stages with telecommands or telecommand sequences         | 64 |
| 3.2.41. | PST  | Command parameter set file which defines the telecommand parameter sets characteristics  | 65 |
| 3.2.42. | PSV  | Command parameter value set file which defines the telecommand parameter value sets characteristics                                    | 65 |
| 3.2.43. | CPS  | Command parameter set file which defines the telecommand parameters contained in a parameter set                                       | 65 |
| 3.2.44. | PVS  | Command parameter value set file which defines the telecommand parameter forming a parameter value set                                 | 65 |



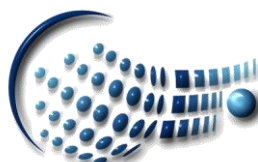


# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|         |     |   |    |
|---------|-----|---|----|
| 3.2.45. | PSM | Command parameter sets mapping file which defines the mapping between parameter sets and tasks (i.e. telecommands or telecommand sequences) | 66 |
| 3.2.46. | CCA | Command parameter calibration curve file which defines the numerical (de-)calibration (for telecommand or telecommand sequence parameters)  | 66 |
| 3.2.47. | CCS | Calibration curve set file which defines the numerical (de-)calibration values (for telecommand or telecommand sequence parameters)         | 67 |
| 3.2.48. | PAF | Command parameter alias file which defines the text (de-)calibration (for telecommand or telecommand sequence parameters)                   | 67 |
| 3.2.49. | PAS | Command parameter alias set which defines the text (de-)calibration values (for telecommand or telecommand sequence parameters)             | 67 |
| 3.2.50. | PRF | Command parameter range file which defines the telecommand command/telecommand sequence parameter range checks                              | 67 |
| 3.2.51. | PRV | Command parameter range value file which defines the telecommand parameter allowed value ranges   | 68 |
| 3.2.52. | GEN | Defines all on-board processors   | 68 |
| 3.2.53. | MAD | Defines all memory areas contained in the on-board processors   | 69 |
| 3.2.54. | MEM | Defines all memory blocks contained in the memory areas   | 69 |
| 3.2.55. | IMG | Defines all memory images contained in the memory areas   | 70 |
| 3.2.56. | CRC | Defines all check sums attached to the memory areas   | 70 |
| 3.2.57. | CAT | Defines all catalogued memory area versions   | 70 |

The following database table descriptions are intended as a quick reference only. Consult the ESA document 'SCOS-2000 Database Import ICD' (EGOS-MCS-S2K-ICD-0001) for a detailed description of this database standard.



### 3.2.1. VDF Table

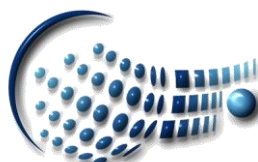
This table defines all database versions.

| No. | Column       | Type     | Description                                  | Restriction      |
|-----|--------------|----------|--|------------------|
| 1   | VDF_NAME     | vc(8)    | Database version name                        | Unique, not null |
| 2   | VDF_COMMENT  | vc(32)   | Comment associated to the database version   | None, null       |
| 3   | VDF_DOMAINID | num(5,0) | Domain identifier associated to the database | 0-65535 or null  |
| 4   | VDF_RELEASE  | num(5,0) | Database release                             | 1-65535 or 0     |
| 5   | VDF_ISSUE    | num(5,0) | Database issue                               | 1-65535 or 0     |

### 3.2.2. PCF Table

This table defines all telemetry parameters.

| No. | Column      | Type      | Description   | Restriction                                |
|-----|-------------|-----------|---|--|
| 1   | PCF_NAME    | vc(8)     | Telemetry parameter name  | Unique, not null                           |
| 2   | PCF_DESCR   | vc(24)    | Textual description of the telemetry parameter  | None, null                                 |
| 3   | PCF_PID     | num(10,0) | On-board ID of the telemetry parameter  | 0-2 <sup>32</sup> -1 or null               |
| 4   | PCF_UNIT    | vc(4)     | Engineering unit mnemonic of the parameter values   | None, null                                 |
| 5   | PCF_PTC     | num(2,0)  | Parameter type code<br><u>Note:</u><br>This is a SCOS-2000 specific number (see Appendix A of the above mentioned document)   | 1-13, not null                             |
| 6   | PCF_PFC     | num(5,0)  | Parameter format code<br><u>Note:</u><br>This is a SCOS-2000 specific number (see Appendix A of the above mentioned document)   | 0-32, not null                             |
| 7   | PCF_WIDTH   | num(6,0)  | 'Padded' width of this telemetry parameter expressed in number of bits<br><u>Note:</u><br>This field reflects this width in bits when the telemetry parameter is extracted from variable length telemetry packets.<br>The default width is derived from the two fields just above (PCF_PTC & PCF_PFC) | 1-32 or null                               |
| 8   | PCF_VALID   | vc(8)     | Validity parameter  | Must refer to a parameter or null          |
| 9   | PCF_RELATED | vc(8)     | Associated parameter  | Must refer to a parameter or null          |
| 10  | PCF_CATEG   | ch(1)     | Calibration category of the parameter:<br>N=Numerical<br>S=Status<br>T=Text (PCF_PTC=8)   | In set {N,S,T}<br>Not null                 |
| 11  | PCF_NATUR   | ch(1)     | Nature of the parameter:<br>R=Raw<br>D=Derived (OL)<br>P=Synthetic (SPEL)<br>H=Hardcoded<br>S=Saved synthetic<br>C=Constant   | In set {R,D,P,H,S,C}<br>Not null           |
| 12  | PCF_CURTX   | vc(10)    | Calibration curve/table name  | Valid calibration curve/table name or null |

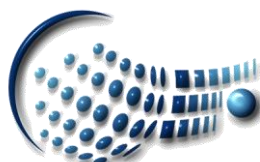


|    |            |          |   |   |
|----|------------|----------|---|---|
| 13 | PCF_INTER  | ch(1)    | Flag controlling the extrapolation behaviour for parameters calibrated using a numerical calibration curve:<br>P=Extrapolated<br>F=Not extrapolated | In set {P,F} or null                    |
| 14 | PCF_USCON  | ch(1)    | Flag controlling the execution of status consistency checks for this parameter:<br>Y=Perform checks<br>N=No checks                                  | In set {Y,N}<br>Not null                |
| 15 | PCF_DECIM  | num(3,0) | Number of decimal places to be used for displaying real values of this monitoring parameter   | 1-16 or null                            |
| 16 | PCF_PARVAL | vc(14)   | Default raw value of the constant parameter   | Valid raw number if PCF_NATUR=C or null |
| 17 | PCF_SUBSYS | vc(8)    | Name of the group to which the monitoring parameter belongs   | None, null                              |
| 18 | PCF_VALPAR | num(5,0) | Raw value of the validity parameter (PCF_VALID)<br><u>Note:</u><br>This is used to evaluate the state validity expression (PCF_VALID=PCF_VALPAR)    | Valid number or null                    |
| 19 | PCF_SPTYPE | ch(1)    | Output view of the synthetic parameter calculation:<br>E=Engineering (calibrated)<br>R=Raw  | In set {E,R} or null                    |
| 20 | PCF_CORR   | ch(1)    | Time parameter correlation flag:<br>Y=Perform correlation<br>N=No correlation   | In set {Y,N}<br>Not null                |
| 21 | PCF_OBTID  | num(5,0) | On-board clock reference  | 1-99999 or 0                            |
| 22 | PCF_DARC   | ch(1)    | Parameter archive flag  | 0 or 1 (not used)<br>Not null           |
| 23 | PCF_ENDIAN | ch(1)    | Parameter endianness:<br>B=Big endian<br>L=Little endian  | In set {B,L}<br>Not null                |

### 3.2.3. CUR Table

This table defines all associations between telemetry parameters and numerical/textual calibration curves.

| No. | Column     | Type     | Description  | Restriction                               |
|-----|------------|----------|--|---|
| 1   | CUR_PNAME  | vc(8)    | Telemetry parameter name   | Valid parameter name<br>Not null          |
| 2   | CUR_POS    | num(2,0) | Evaluation sequence number   | 0-99<br>Not null                          |
| 3   | CUR_RLCHK  | vc(8)    | Validity parameter   | Valid parameter name or null              |
| 4   | CUR_VALPAR | num(5,0) | Raw value of the validity parameter<br><u>Note:</u><br>This is used to evaluate the calibration validity expression (CUR_RLCHK=CUR_VALPAR) | Valid raw number if CUR_RLCHK is not null |
| 5   | CUR_SELECT | vc(10)   | Calibration curve name   | Valid calibration curve name<br>Not null  |



### 3.2.4. CAF Table

This table defines all numerical telemetry calibration curves.

| No. | Column     | Type     | Description   | Restriction                       |
|-----|------------|----------|---|-----------------------------------|
| 1   | CAF_NUMBR  | vc(10)   | Numerical calibration curve name  | Unique, not null                  |
| 2   | CAF_DESCR  | vc(32)   | Description of the calibration curve  | None, null                        |
| 3   | CAF_ENGFMT | ch(1)    | Engineering format type flag:<br>I = Signed integer<br>U = Unsigned integer<br>R = Floating-point | In set {I,U,R}<br>Not null        |
| 4   | CAF_RAWFMT | ch(1)    | Raw format type flag:<br>I = Signed integer<br>U = Unsigned integer<br>R = Floating-point         | In set {I,U,R}<br>Not null        |
| 5   | CAF_RADIX  | ch(1)    | Radix for the raw format type:<br>D = Decimal<br>H = Hexadecimal<br>O = Octal                     | In set {D,H,O} if<br>CAF_RAWFMT=U |
| 6   | CAF_UNIT   | vc(4)    | Engineering unit mnemonic   | None, null                        |
| 7   | CAF_NCURVE | num(3,0) | Number of points defined for this calibration curve   | 0-999<br>Not null                 |
| 8   | CAF_INTER  | ch(1)    | Extrapolation flag:<br>P = Extrapolated<br>F = Not extrapolated                                   | In set {P,F}<br>Not null          |

### 3.2.5. CAP Table

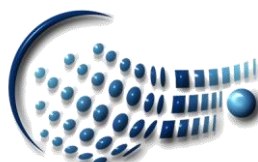
This table defines all points allocated to numerical telemetry calibration curves.

| No. | Column    | Type   | Description  | Restriction                              |
|-----|-----------|--------|--|--|
| 1   | CAP_NUMBR | vc(10) | Numerical calibration curve name   | Valid calibration curve name<br>Not null |
| 2   | CAP_XVALS | vc(14) | Raw value of the calibration point<br><b>Note:</b><br>This value has to be expressed in a format compatible with CAF_RAWFMT; unsigned integer values have to be expressed using the radix associated to this calibration curve (CAF_RADIX)               | Valid raw number<br>Not null             |
| 3   | CAP_YVALS | vc(14) | Engineering value of the calibration point<br><b>Note:</b><br>This value has to be expressed in a format compatible with CAF_ENGFMT; all values have to be expressed in decimal regardless of the radix associated to this calibration curve (CAF_RADIX) | Valid engineering number<br>Not null     |

### 3.2.6. TXF Table

This table defines all textual telemetry calibration curves.

| No. | Column    | Type   | Description                          | Restriction      |
|-----|-----------|--------|--------------------------------------|------------------|
| 1   | TXF_NUMBR | vc(10) | Textual calibration table name       | Unique, not null |
| 2   | TXF_DESCR | vc(32) | Description of the calibration table | None, null       |





# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|   |            |          |   |                            |
|---|------------|----------|---|----------------------------|
| 3 | TXF_RAWFMT | ch(1)    | Raw format type flag:<br>I = Signed integer<br>U = Unsigned integer<br>R = Floating-point | In set {I,U,R}<br>Not null |
| 4 | TXF_NALIAS | num(3,0) | Number of entries defined for this calibration table                                      | 0-999<br>Not null          |

### 3.2.7. TXP Table

This table defines all entries allocated to textual telemetry calibration tables.

| No. | Column    | Type   | Description  | Restriction                              |
|-----|-----------|--------|--|--|
| 1   | TXP_NUMBR | vc(10) | Textual calibration table name   | Valid calibration table name<br>Not null |
| 2   | TXP_FROM  | vc(14) | Lower range raw value<br><u>Note:</u><br>This value has to be expressed in a format compatible with TXF_RAWFMT (decimal radix) | Valid raw number<br>Not null             |
| 3   | TXP_TO    | vc(14) | Upper range raw value<br><u>Note:</u><br>This value has to be expressed in a format compatible with TXF_RAWFMT (decimal radix) | Valid raw number<br>Not null             |
| 4   | TXP_ALTXT | vc(14) | Engineering value text   | None, not null                           |

### 3.2.8. MCF Table

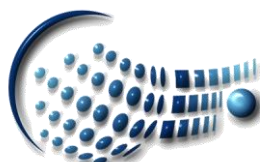
This table defines all polynomial telemetry calibration curves.

| No. | Column    | Type   | Description   | Restriction            |
|-----|-----------|--------|---|------------------------|
| 1   | MCF_IDENT | vc(10) | Polynomial calibration curve name                             | Unique, not null       |
| 2   | MCF_DESCR | vc(32) | Description of the calibration curve                          | None, null             |
| 3   | MCF_POL1  | vc(14) | Polynomial coefficient $A_0$                                  | Valid number, not null |
| 4   | MCF_POL2  | vc(14) | Polynomial coefficient of the 1 <sup>st</sup> order ( $A_1$ ) | Valid number or 0      |
| 5   | MCF_POL3  | vc(14) | Polynomial coefficient of the 2 <sup>nd</sup> order ( $A_2$ ) | Valid number or 0      |
| 6   | MCF_POL4  | vc(14) | Polynomial coefficient of the 3 <sup>rd</sup> order ( $A_3$ ) | Valid number or 0      |
| 7   | MCF_POL5  | vc(14) | Polynomial coefficient of the 4 <sup>th</sup> order ( $A_4$ ) | Valid number or 0      |

### 3.2.9. LGF Table

This table defines all logarithmic telemetry calibration curves.

| No. | Column    | Type   | Description  | Restriction            |
|-----|-----------|--------|--|------------------------|
| 1   | LGF_IDENT | vc(10) | Logarithmic calibration curve name                             | Unique, not null       |
| 2   | LGF_DESCR | vc(32) | Description of the calibration curve                           | None, null             |
| 3   | LGF_POL1  | vc(14) | Logarithmic coefficient $A_0$                                  | Valid number, not null |
| 4   | LGF_POL2  | vc(14) | Logarithmic coefficient of the 1 <sup>st</sup> order ( $A_1$ ) | Valid number or 0      |
| 5   | LGF_POL3  | vc(14) | Logarithmic coefficient of the 2 <sup>nd</sup> order ( $A_2$ ) | Valid number or 0      |
| 6   | LGF_POL4  | vc(14) | Logarithmic coefficient of the 3 <sup>rd</sup> order ( $A_3$ ) | Valid number or 0      |
| 7   | LGF_POL5  | vc(14) | Logarithmic coefficient of the 4 <sup>th</sup> order ( $A_4$ ) | Valid number or 0      |



### 3.2.10. OCF Table

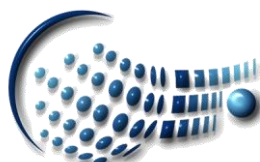
This table defines the numerical/textual out-of-limit tables associated to the telemetry parameters.

| No. | Column     | Type     | Description  | Restriction                      |
|-----|------------|----------|--|----------------------------------|
| 1   | OCF_NAME   | vc(8)    | Telemetry parameter name   | Valid parameter name<br>Not null |
| 2   | OCF_NBCHCK | num(2,0) | Threshold number<br><u>Note:</u><br>This number indicates how many consecutive valid parameter samples violating the check are required to declare the parameter as out-of-limit | 1-99<br>Not null                 |
| 3   | OCF_NBOOL  | num(2,0) | Number of checks   | 1-16<br>Not null                 |
| 4   | OCF_INTER  | ch(1)    | Limit values interpretation flag:<br>U=Raw values<br>C=Calibrated values   | In set {U,C}<br>Not null         |
| 5   | OCF_CODIN  | ch(1)    | Limit values coding flag:<br>I =Signed integer<br>R=Floating-point<br>A=Status text  | In set {I,R,A}<br>Not null       |

### 3.2.11. OCP Table

This table defines all value limit checks associated to the telemetry out-of-limit tables.

| No. | Column    | Type     | Description   | Restriction                      |
|-----|-----------|----------|---|----------------------------------|
| 1   | OCP_NAME  | vc(8)    | Telemetry parameter name  | Valid parameter name<br>Not null |
| 2   | OCP_POS   | num(2,0) | Evaluation sequence number<br><u>Note:</u><br>The checks need to be sorted with the pairs ordered by soft and hard entries  | 1-99<br>Not null                 |
| 3   | OCP_TYPE  | ch(1)    | Check type flag:<br>S =Soft limit check<br>H=Hard limit or soft status check<br>D=Delta check<br>C=Status consistency<br>E =Event generation only (no OOL)  | In set {S,H,D,C,E}<br>Not null   |
| 4   | OCP_LVALU | vc(14)   | Low limit value<br><u>Note:</u><br>- For delta checks this field specifies the minimum (allowed) delta limit value<br>- This field is irrelevant for status consistency checks<br>- If specified, this value must be expressed as specified by OCF_INTER/OCF_CODIN                    | Valid limit value or null        |
| 5   | OCP_HVALU | vc(14)   | High limit value<br><u>Note:</u><br>- For delta checks this field specifies the maximum (allowed) delta limit value<br>- This field is irrelevant for status checks and status consistency checks<br>- If specified, this value must be expressed as specified by OCF_INTER/OCF_CODIN | Valid limit value or null        |

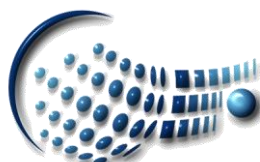


|   |            |          |  |   |
|---|------------|----------|--|---|
| 6 | OCP_RLCHK  | vc(8)    | Validity parameter<br><u>Note:</u><br>This field is irrelevant for status consistency checks   | Valid parameter name or null              |
| 7 | OCP_VALPAR | num(5,0) | Raw value of the validity parameter<br><u>Note:</u><br>- This is used to evaluate the calibration validity expression (OCP_RLCHK=OCP_VALPAR)<br>- This field is irrelevant for status consistency checks | Valid raw number if OCP_RLCHK is not null |

### 3.2.12. PID Table

This table defines the identification details for all telemetry packets.

| No. | Column      | Type      | Description   | Restriction  |
|-----|-------------|-----------|---|--|
| 1   | PID_TYPE    | num(3,0)  | Telemetry source packet type  | 0-255<br>Not null  |
| 2   | PID_STYPE   | num(3,0)  | Telemetry source packet sub-type  | 0-255<br>Not null  |
| 3   | PID_APID    | num(5,0)  | Telemetry source packet application identifier  | 0-2047 for spacecraft packets<br>2048-65535 for others<br>Not null |
| 4   | PID_PI1_VAL | num(10,0) | Value of the first telemetry source packet additional identification field<br><u>Note:</u><br>This field is used, together with the packet application identifier, type and sub-type to identify the packet structure and to interpret its content  | Valid number or 0  |
| 5   | PID_PI2_VAL | num(10,0) | Value of the second telemetry source packet additional identification field<br><u>Note:</u><br>This field is used, together with the packet application identifier, type and sub-type to identify the packet structure and to interpret its content | Valid number or 0  |
| 6   | PID_SPID    | num(10,0) | Telemetry source packet number<br><u>Note:</u><br>This field uniquely identifies the structure of packets   | Valid number<br>Not null   |
| 7   | PID_DESCR   | vc(64)    | Description of the packet   | None, null   |
| 8   | PID_UNIT    | vc(8)     | On-board sub-system   | None, null   |
| 9   | PID_TPSD    | num(10,0) | Packet structure definition<br><u>Note:</u><br>This field has to be set for variable size packets   | Valid number or -1   |
| 10  | PID_DFHSIZE | num(2,0)  | Data field header size<br><u>Note:</u><br>This field is irrelevant in case of fixed size packets (PID_TPSD=-1)  | 0-99<br>Not null   |
| 11  | PID_TIME    | ch(1)     | Time field flag:<br>Y = This field is present<br>N = No time field  | In set {Y,N}<br>Not null   |



# BINARY SPACE

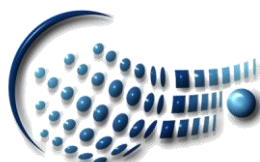
## RELIABLE SPACE SYSTEMS

|    |           |          |   |                              |
|----|-----------|----------|---|------------------------------|
| 12 | PID_INTER | num(8,0) | Packet generation interval (in ms)<br><u>Note:</u><br>This field is used for periodic telemetry packets only  | Valid number or null         |
| 13 | PID_VALID | ch(1)    | Flag indicating whether the combination of the key fields shall be considered as valid:<br>Y =The key combination is valid<br>N=Not valid<br><u>Note:</u><br>This field can be used to identify if the PID_SPID field can be considered as additional key in case more records with the same key fields exist | In set {Y,N}<br>Not null     |
| 14 | PID_CHECK | num(1,0) | Packet error control flag:<br>0=No packet error control field present (or no check required)<br>1=Packet error control field present  | In set {0,1}<br>Not null     |
| 15 | PID_EVENT | ch(1)    | Event message generation flag:<br>N=No event message generation<br>I =Generation of an informational event message<br>W=Generation of a warning event message<br>A =Generation of an error event message  | In set {N,I,W,A}<br>Not null |
| 16 | PID_EVID  | vc(17)   | Event message text  | None, null if PID_EVENT=N    |

### 3.2.13. PIC Table

This table defines additional telemetry packet identification details including positional information.

| No. | Column      | Type     | Description  | Restriction                                     |
|-----|-------------|----------|--|---|
| 1   | PIC_TYPE    | num(3,0) | Telemetry source packet type   | 0-255<br>Not null                               |
| 2   | PIC_STYPE   | num(3,0) | Telemetry source packet sub-type   | 0-255<br>Not null                               |
| 3   | PIC_PI1_OFF | num(5,0) | Offset of the first packet identification field (in bytes)   | 0-65535 or -1<br>Not null                       |
| 4   | PIC_PI1_WID | num(3,0) | Width of the first additional packet identification field (in bits)  | 1-32<br>Not null                                |
| 5   | PIC_PI2_OFF | num(5,0) | Offset of the second packet identification field (in bytes)  | 0-65535 or -1<br>Not null                       |
| 6   | PIC_PI2_WID | num(3,0) | Width of the second additional packet identification field (in bits)   | 1-32<br>Not null                                |
| 7   | PIC_APID    | num(5,0) | Application identifier of the packet<br><u>Note:</u><br>If this field is omitted or left blank, no application identifier value is specified for this type/sub-type combination and thus this record definition is valid for all packets | Valid application identification number or null |





### 3.2.14. TPCF Table

This table defines the attributes for all telemetry packets.

| No. | Column    | Type      | Description  | Restriction      |
|-----|-----------|-----------|--|------------------|
| 1   | TPCF_SPID | num(10,0) | Telemetry source packet number   | Unique, not null |
| 2   | TPCF_NAME | vc(12)    | Packet mnemonic  | None, null       |
| 3   | TPCF_SIZE | num(8,0)  | Total packet size (in bytes)<br>Note:<br>This field has to be set to 0 for variable size packets | 1-65535 or 0     |

### 3.2.15. PLF Table

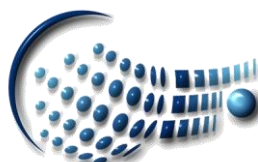
This table defines the parameter locations for all fixed size telemetry packets.

| No. | Column    | Type      | Description  | Restriction                      |
|-----|-----------|-----------|--|----------------------------------|
| 1   | PLF_NAME  | vc(8)     | Name of the telemetry parameter  | Valid parameter name<br>Not null |
| 2   | PLF_SPID  | num(10,0) | Telemetry source packet number   | Valid packet number<br>Not null  |
| 3   | PLF_OFFBY | num(5,0)  | Location of first occurrence of the specified parameter's value (in bytes)                     | 0-65535<br>Not null              |
| 4   | PLF_OFFBI | num(1,0)  | Bit position, within a byte, of the first bit of the first occurrence of the parameter's value | 0-7<br>Not null                  |
| 5   | PLF_NBOCC | num(4,0)  | Number of occurrences of the parameter in the packet   | 1-9999<br>Not null               |
| 6   | PLF_LGOCC | num(5,0)  | Number of bits between the start of two consecutive parameter occurrences                      | 0-32767<br>Not null              |
| 7   | PLF_TIME  | num(9,0)  | Time offset of first parameter occurrence relative to packet time (in ms)                      | -4080000-4080000<br>Not null     |
| 8   | PLF_TDOCC | num(9,0)  | Time delay between two consecutive parameter occurrences (in ms)                               | 1-4080000<br>Not null            |

### 3.2.16. VPD Table

This table defines the parameter locations for all variable size telemetry packets.

| No. | Column      | Type      | Description  | Restriction                      |
|-----|-------------|-----------|--|----------------------------------|
| 1   | VPD_TPSD    | num(10,0) | Telemetry source packet structure definition   | Valid number<br>Not null         |
| 2   | VPD_POS     | num(4,0)  | Ordinal position of this parameter inside the telemetry source packet definition   | 0-9999<br>Not null               |
| 3   | VPD_NAME    | vc(8)     | Name of the telemetry parameter  | Valid parameter name<br>Not null |
| 4   | VPD_GRPSize | num(3,0)  | Group size (in parameters)   | 1-999 or 0                       |
| 5   | VPD_FIXREP  | num(3,0)  | Group repeater count   | 1-999 if VPD_GRPSize>0<br>or 0   |
| 6   | VPD_CHOICE  | ch(1)     | Flag identifying whether this parameter's value is to be used to determine the structure of the following portion of the packet:<br>Y =Used as key (VPD_TPSD) to interpret the following entries in the packet<br>N=No choice type | In set {Y,N}<br>Not null         |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |             |          |   |                                |
|----|-------------|----------|---|--------------------------------|
| 7  | VPD_PIDREF  | ch(1)    | Flag indicating whether this parameter's value provides the on-board identifier of a telemetry parameter:<br>Y =Used as a parameter on-board identifier (PCF_PID)<br>N=No on-board identifier   | In set {Y,N}<br>Not null       |
| 8  | VPD_DISDESC | vc(16)   | Textual description of the parameter value  | None, null                     |
| 9  | VPD_WIDTH   | num(2,0) | Field width used to control whether the parameter value is displayed:<br>0 =Value is not displayed<br>>0=Value is displayed   | 0-99<br>Not null               |
| 10 | VPD_JUSTIFY | ch(1)    | Flag controlling the justification used in the display for the value field:<br>L =Left aligned<br>C=Centered<br>R =Right aligned  | In set {L,C,R}<br>Not null     |
| 11 | VPD_NEWLINE | ch(1)    | Flag controlling whether the parameter shall be displayed on a new line:<br>Y =New line<br>N=No new line  | In set {Y,N}<br>Not null       |
| 12 | VPD_DCHAR   | num(1,0) | Flag controlling which data are displayed for the parameter:<br>0=Only the value is displayed<br>1=Parameter name and value are displayed<br>2=Parameter name, description and value are displayed  | 0-2<br>Not null                |
| 13 | VPD_FORM    | ch(1)    | Display format of the parameter's value:<br>N=Normal (as specified in the parameter definition)<br>B =Binary<br>O=Octal<br>D=Decimal<br>H =Hexadecimal<br><u>Note:</u><br>The format codes {B,O,D,H} apply for raw values of parameters with an unsigned integer coding only      | In set {N,B,O,D,H}<br>Not null |
| 14 | VPD_OFFSET  | num(6,0) | Number of bits between the start position of this parameter and the end bit of the previous parameter in the packet<br><u>Note:</u><br>A negative offset enables the 'overlap' of the bits contributing to this parameter with the ones contributing to the previous parameter(s) | -32767-32767<br>Not null       |



### 3.2.17. GRP Table

This table defines all telemetry packet and telemetry parameter groups.

| No. | Column    | Type   | Description   | Restriction                |
|-----|-----------|--------|---|----------------------------|
| 1   | GRP_NAME  | vc(14) | Telemetry monitoring group name   | Unique, not null           |
| 2   | GRP_DESCR | vc(24) | Textural description of the monitoring group  | None, not null             |
| 3   | GRP_GTYPE | ch(2)  | Type of the monitoring group:<br>PA=Telemetry parameter group<br>PK=Telemetry source packet group | In set {PA,PK}<br>Not null |

### 3.2.18. GRPA Table

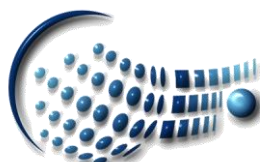
This table defines the content of all telemetry parameter groups.

| No. | Column      | Type   | Description  | Restriction                      |
|-----|-------------|--------|--|----------------------------------|
| 1   | GRPA_GNAME  | vc(14) | Name of the monitoring group to which this telemetry parameter belongs   | Valid group name<br>Not null     |
| 2   | GRPA_PANAME | vc(8)  | Name of the telemetry parameter belonging to this group<br><u>Note:</u><br>Only monitoring parameters that are part of a telemetry packet or those derived from such parameters can be specified | Valid parameter name<br>Not null |

### 3.2.19. GRPK Table

This table defines the content of all telemetry packet groups.

| No. | Column      | Type      | Description  | Restriction                     |
|-----|-------------|-----------|--|---------------------------------|
| 1   | GRPK_GNAME  | vc(14)    | Name of the monitoring group to which this telemetry packet belongs  | Valid group name<br>Not null    |
| 2   | GRPK_PKSPID | num(10,0) | Packet number of the telemetry source packet belonging to this group | Valid packet number<br>Not null |



### 3.2.20. DPF Table

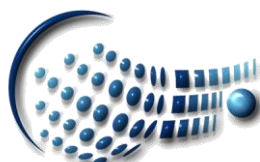
This table defines all alphanumeric displays (ANDs).

| No. | Column    | Type   | Description   | Restriction              |
|-----|-----------|--------|---|--------------------------|
| 1   | DPF_NUMBE | vc(8)  | Name of the alphanumeric display (AND)  | Unique, not null         |
| 2   | DPF_TYPE  | ch(1)  | Type of the alphanumeric display:<br>1=32-parameters display<br>3=64-parameters display | In set {1,3}<br>Not null |
| 3   | DPF_HEAD  | vc(32) | Textual description of the alphanumeric display   | None, null               |

### 3.2.21. DPC Table

This table defines the content of all alphanumeric displays (ANDs).

| No. | Column    | Type     | Description   | Restriction  |
|-----|-----------|----------|---|--|
| 1   | DPC_NUMBE | vc(8)    | Name of the alphanumeric display (AND)  | Valid display name<br>Not null                       |
| 2   | DPC_NAME  | vc(8)    | Name of the telemetry parameter belonging to this display<br><u>Note:</u><br>If this field is left empty the DPC_TEXT field must contain a text (e.g. group title)  | Valid parameter name or null                         |
| 3   | DPC_FLDN  | num(2,0) | Parameter position in the display   | 0-31 =Left column<br>32-63 =Right column<br>Not null |
| 4   | DPC_COMM  | num(4,0) | Commutation of the parameter to be displayed  | 0-9999<br>Not null                                   |
| 5   | DPC_MODE  | ch(1)    | Flag enabling/disabling the visibility of invalid parameter values:<br>Y =Invalid values are not displayed<br>N=Values are always displayed   | In set {Y,N}<br>Not null                             |
| 6   | DPC_FORM  | ch(1)    | Display format of the parameter's value:<br>N=Normal (as specified in the parameter definition)<br>B =Binary<br>O=Octal<br>D=Decimal<br>H=Hexadecimal<br><u>Note:</u><br>The format codes {B,O,D,H} apply for raw values of parameters with an unsigned integer coding only | In set {N,B,O,D,H}<br>Not null                       |
| 7   | DPC_TEXT  | vc(32)   | Text comment  | None, null if DPC_NAME is not null                   |





# BINARY SPACE

## RELIABLE SPACE SYSTEMS

Satview™ also supports the definition of alphanumeric displays via XML files.  
The following schema definition is used:

```
<?xml version="1.0" encoding="utf-8" ?>
<!ATTLIST ANDdef
    name CDATA #REQUIRED
    description CDATA #REQUIRED>
<!ELEMENT ANDdef (ANDparamDef*, CommentLine*)>
<!ELEMENT ANDparamDef EMPTY>
<!ATTLIST ANDparamDef
    name CDATA #REQUIRED
    domain CDATA #OPTIONAL
    representation (SOURCE | CALIBRATED | DEFAULT) DEFAULT
    orderInDisplay CDATA #REQUIRED
    commutation CDATA 0
    format (BINARY | OCTAL | DECIMAL | HEXADECIMAL | NORMAL) NORMAL
    displayIfInvalid (true | false) false>
<!ELEMENT CommentLine EMPTY>
<!ATTLIST CommentLine
    orderInDisplay CDATA #REQUIRED
    comment CDATA #REQUIRED>
```

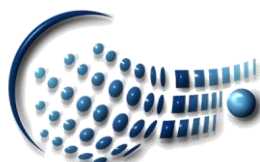
### Example:

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE ANDdef SYSTEM 'ANDdef.dtd'>
<ANDdef name="S2KAND01" description="32 Parameter Display">
<ANDparamDef name="S2KTP301" orderInDisplay="0" commutation="0" format="BINARY"
displayIfInvalid="true"/>
<ANDparamDef name="S2KTP302" orderInDisplay="8" commutation="1" format="OCTAL"
displayIfInvalid="true"/>
<ANDparamDef name="S2KTP303" orderInDisplay="16" commutation="1" format="DECIMAL"
displayIfInvalid="true"/>
<ANDparamDef name="S2KTP304" orderInDisplay="24" commutation="1" format="HEXADECIMAL"
displayIfInvalid="true"/>
<ANDparamDef name="S2KTP305" orderInDisplay="31" commutation="1" format="NORMAL"
displayIfInvalid="true"/>
<CommentLine orderInDisplay="1" comment="This is 1st parameter"/>
<CommentLine orderInDisplay="7" comment="This is 2nd parameter "/>
<CommentLine orderInDisplay="25" comment="This is 3rd parameter "/>
<CommentLine orderInDisplay="30" comment="This is 4th parameter "/>
</ANDdef>
```

The mapping between DBMS table and XML files is as follows:

| Field Name | ANDDef Attribute |
|------------|------------------|
| DPF_NUMBE  | name             |
| DPF_TYPE   | -                |
| DPF_HEAD   | description      |

| Field Name | ANDparamDef Attribute |
|------------|-----------------------|
| DPC_NUMBE  | -                     |
| DPC_NAME   | name                  |
| DPC_FLDN   | orderInDisplay        |
| DPC_COMM   | commutation           |
| DPC_MODE   | displayIfInvalid      |
| DPC_FORM   | format                |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

### 3.2.22. GPF Table

This table defines all graphic displays (GRDs).

| No. | Column    | Type     | Description   | Restriction                         |
|-----|-----------|----------|---|-------------------------------------|
| 1   | GPF_NUMBE | vc(8)    | Name of the graphic display (GRD)   | Unique, not null                    |
| 2   | GPF_TYPE  | ch(1)    | Type of the graphic display:<br>F =Plots against time or another parameter<br>H =Strip chart plot<br>Q=Strip chart plot<br>S =Strip chart plot<br>Note:<br>The type codes {H,Q,S} have the same meaning | In set {F,H,Q,S}<br>Not null        |
| 3   | GPF_HEAD  | vc(32)   | Textual description of the graphic display  | None, null                          |
| 4   | GPF_SCROL | ch(1)    | Flag controlling the scrolling behavior of the display when reaching the end of the plot area:<br>Y =Scrolling is enabled<br>N=Scrolling is disabled (i.e. the plot is overwritten)                     | In set {Y,N}<br>Not null            |
| 5   | GPF_HCOPY | ch(1)    | Flag controlling the auto hardcopy behavior at the end of plot area:<br>Y =Perform auto hardcopy<br>N=No auto hardcopy  | In set {Y,N} (not used)<br>Not null |
| 6   | GPF_DAYS  | num(2,0) | Plot duration (in days)   | 0-99<br>Not null                    |
| 7   | GPF_HOURS | num(2,0) | Plot duration (in hours)  | 0-23<br>Not null                    |
| 8   | GPF_MINUT | num(2,0) | Plot duration (in minutes)  | 0-59<br>Not null                    |
| 9   | GPF_AXCLR | ch(1)    | Color of axes, ticks and grids:<br>1 =Green<br>2 =Blue<br>3 =Cyan<br>4 =Red<br>5 =Yellow<br>6 =Magenta<br>7 =Black  | In set {1,2,3,4,5,6,7}<br>Not null  |
| 10  | GPF_XTICK | num(2,0) | Number of intervals between ticks on X-axis   | 1-99<br>Not null                    |
| 11  | GPF_YTICK | num(2,0) | Number of intervals between ticks on Y-axis   | 1-99<br>Not null                    |
| 12  | GPF_XGRID | num(2,0) | Number of intervals between grids on X-axis   | 1-99<br>Not null                    |
| 13  | GPF_YGRID | num(2,0) | Number of intervals between grids on Y-axis   | 1-99<br>Not null                    |
| 14  | GPF_UPUN  | num(2,0) | Display update rate (in seconds)  | 1-99 (not used) or null             |



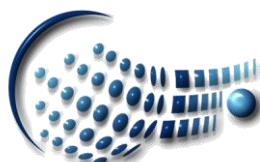
# BINARY SPACE

## RELIABLE SPACE SYSTEMS

### 3.2.23. GPC Table

This table defines the content of all graphic displays (GRDs).

| No. | Column     | Type     | Description  | Restriction                        |
|-----|------------|----------|--|------------------------------------|
| 1   | GPC_NUMBE  | vc(8)    | Name of the graphic display (GRD)  | Valid display name<br>Not null     |
| 2   | GPC_POS    | num(1,0) | Parameter position in the display  | 0-7<br>Not null                    |
| 3   | GPC_WHERE  | ch(1)    | Identification of the plot area where the parameter is to be plotted:<br>1-8 = On Y-axis (from top to bottom)<br>P = On X-axis | 1-8 or P<br>Not null               |
| 4   | GPC_NAME   | vc(8)    | Name of the telemetry parameter belonging to this display  | Valid parameter name<br>Not null   |
| 5   | GPC_RAW    | ch(1)    | Flag controlling whether the calibrated or raw value shall be plotted:<br>C=Calibrated value<br>U=Raw value                    | In set {U,C}<br>Not null           |
| 6   | GPC_MINIM  | vc(14)   | Minimum value for the parameter<br><u>Note:</u><br>The value needs to be expressed as specified by GPC_RAW                     | Valid number<br>Not null           |
| 7   | GPC_MAXIM  | vc(14)   | Maximum value for the parameter<br><u>Note:</u><br>The value needs to be expressed as specified by GPC_RAW                     | Valid number<br>Not null           |
| 8   | GPC_PRCLR  | ch(1)    | Plot line and symbol colors:<br>1=Green<br>2=Blue<br>3=Cyan<br>4=Red<br>5=Yellow<br>6=Magenta<br>7=Black                       | In set {1,2,3,4,5,6,7}<br>Not null |
| 9   | GPC_SYMB0  | ch(1)    | Plot symbol:<br>0 =No symbol<br>1 =Cross<br>2 =Stroke<br>3 =Point<br>4-6 =Square   | 0-6<br>Not null                    |
| 10  | GPC_LINE   | ch(1)    | Plot line type:<br>1 =Solid line<br>2-3 =Dashed line<br>4-5 =Dotted line   | 1-5 or 0<br>Not null               |
| 11  | GPC_DOMAIN | num(5,0) | Domain identifier to which the monitoring parameter belongs to   | 0-65535 (not used) or null         |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

Satview™ also supports the definition of graphic displays via XML files:

```
<?xml version="1.0" encoding="utf-8" ?>
<!ATTLIST GRDdef
  name CDATA #REQUIRED
  description CDATA #REQUIRED
  plotType (FULL_SCREEN | HALF_SCREEN | QUATER_SCREEN | SLICED_SCREEN) #REQUIRED
  isScrolling (true | false) false
  autoHardCopy (true | false) false
  duration CDATA #REQUIRED
  color (WHITE | GREEN | BLUE | CYAN | MAGENTA | RED | YELLOW) #REQUIRED
  xAxisPoints CDATA #REQUIRED
  yAxisPoints CDATA #REQUIRED
  xGridLines CDATA #REQUIRED
  yGridLines CDATA #REQUIRED>
<!ELEMENT GRDdef (GRDparamDef*, CommentLine*)>
<!ELEMENT GRDparamDef EMPTY>
<!ATTLIST GRDparamDef
  name CDATA #REQUIRED
  domain CDATA #OPTIONAL
  representation (SOURCE | CALIBRATED | DEFAULT) DEFAULT
  orderInDisplay CDATA #REQUIRED
  initWithSourceView (true | false) true
  plotArea CDATA #REQUIRED
  isXaxisParameter (true | false) true
  minValue CDATA #REQUIRED
  maxValue CDATA #REQUIRED
  color (WHITE | GREEN | BLUE | CYAN | MAGENTA | RED | YELLOW) #REQUIRED
  marker (POINT | CROSS | PLUS | CIRCLE | STAR | DIAMOND | SQUARE) POINT
  lineStyle (NONE | SOLID | LONGDASH | SHORTDASH | DOT | DASHDOT) NONE>
```

Example:

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE GRDdef SYSTEM 'GRDdef.dtd'>
<GRDdef name="S2KGRD01" description="Full Green Scrolling" plotType="FULL_SCREEN" isScrolling="true"
autoHardCopy="false" duration="5" color="GREEN" xAxisPoints="10" yAxisPoints="1" xGridLines="1"
yGridLines="1">
<GRDparamDef name="S2KTP306" orderInDisplay="0" initWithSourceView="true" plotArea="0"
isXaxisParameter="false" minValue="0" maxValue="1024" color="WHITE" marker="POINT"
lineStyle="DASHDOT"/>
</GRDdef>
```

The mapping between DBMS table and XML files is as follows:

| Field Name | GRDDef Attribute | Field Name | GRDparamDef Attribute |
|------------|------------------|------------|-----------------------|
| GPf_NUMBE  | name             | GPC_NUMBE  | -                     |
| GPf_TYPE   | plotType         | GPC_POS    | orderInDisplay        |
| GPf_HEAD   | description      | GPC_WHERE  | plotArea              |
| GPf_SCROL  | isScrolling      | GPC_NAME   | name                  |
| GPf_HCOPY  | autoHardCopy     | GPC_RAW    | initWithSourceView    |
| GPf_DAYS   | duration         | GPC_MINIM  | minValue              |
| GPf_HOURS  | duration         | GPC_MAXIM  | maxValue              |
| GPf_MINUT  | duration         | GPC_PRCLR  | color                 |
| GPf_AXCLR  | color            | GPC_SYMBO  | marker                |
| GPf_XTICK  | xAxisPoints      | GPC_LINE   | lineStyle             |
| GPf_YTICK  | yAxisPoints      |            |                       |
| GPf_XGRID  | xGridLines       |            |                       |
| GPf_YGRID  | yGridLines       |            |                       |
| GPf_UPUN   | -                |            |                       |





### 3.2.24. SPF Table

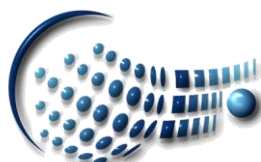
This table defines all parameter observation displays (PODs).

| No. | Column    | Type     | Description  | Restriction             |
|-----|-----------|----------|--|-------------------------|
| 1   | SPF_NUMBE | vc(8)    | Name of the parameter observation display (POD)          | Unique, not null        |
| 2   | SPF_HEAD  | vc(32)   | Textual description of the parameter observation display | None, null              |
| 3   | SPF_NPAR  | num(1,0) | The number of parameters in the display                  | 1-5<br>Not null         |
| 4   | SPF_UPUN  | num(2,0) | Display update rate in seconds                           | 1-99 (not used) or null |

### 3.2.25. SPC Table

This table defines the content of all parameter observation displays (PODs).

| No. | Column    | Type     | Description   | Restriction                      |
|-----|-----------|----------|---|----------------------------------|
| 1   | SPC_NUMBE | vc(8)    | Name of the parameter observation display (POD)   | Valid display name<br>Not null   |
| 2   | SPC_POS   | num(1,0) | Parameter position in the display   | 0-4<br>Not null                  |
| 3   | SPC_NAME  | vc(8)    | Name of the telemetry parameter belonging to this display   | Valid parameter name<br>Not null |
| 4   | SPC_UPDT  | ch(1)    | Flag controlling whether the parameter should be listed in the scrollable area or not:<br>' ' =List in the scrollable area<br>N =List in the over-writable area   | In set {' ',N}<br>Not null       |
| 5   | SPC_MODE  | ch(1)    | Flag enabling/disabling the visibility of invalid parameter values:<br>' ' =Invalid values are not displayed<br>N =Values are always displayed  | In set {' ',N}<br>Not null       |
| 6   | SPC_FORM  | ch(1)    | Display format of the parameter's value:<br>N=Normal (as specified in the parameter definition)<br>B =Binary<br>O=Octal<br>D=Decimal<br>H=Hexadecimal<br><u>Note:</u><br>The format codes {B,O,D,H} apply for raw values of parameters with an unsigned integer coding only | In set {N,B,O,D,H}<br>Not null   |
| 7   | SPC_BACK  | ch(1)    | Background color field:<br>0=Black<br>1=Green<br>2=Blue<br>3=Cyan<br>4=Red<br>5=Yellow<br>6=Magenta<br>7=White  | 0-7<br>Not null                  |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|   |          |       |  |                 |
|---|----------|-------|--|-----------------|
| 8 | SPC_FORE | ch(1) | Foreground color field:<br>0=Black<br>1=Green<br>2=Blue<br>3=Cyan<br>4=Red<br>5=Yellow<br>6=Magenta<br>7=White | 0-7<br>Not null |
|---|----------|-------|--|-----------------|

Satview™ also supports the definition of parameter observation displays via XML files.  
The following schema definition is used:

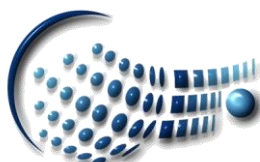
```
<?xml version="1.0" encoding="utf-8" ?>
<!ATTLIST SCDdef
    name CDATA #REQUIRED
    description CDATA #REQUIRED>
<!ELEMENT SCDdef (SCDparamDef*, CommentLine*)>
<!ELEMENT SCDparamDef EMPTY>
<!ATTLIST SCDparamDef
    name CDATA #REQUIRED
    domain CDATA #OPTIONAL
    representation (SOURCE | CALIBRATED | DEFAULT) DEFAULT
    orderInDisplay CDATA #REQUIRED
    format (BINARY | OCTAL | DECIMAL | HEXADECIMAL | NORMAL) NORMAL
    displayIfInvalid (true | false) false
    fgColor (WHITE | BLACK | GREEN | BLUE | CYAN | MAGENTA | RED | YELLOW) #REQUIRED
    bgColor (WHITE | BLACK | GREEN | BLUE | CYAN | MAGENTA | RED | YELLOW) BLACK
    updateWhenScroll (true | false) true>
```

### Example:

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE SCDdef SYSTEM 'SCDdef.dtd'>
<SCDdef name="S2KSCD01" description="S2KSCD01">
<SCDparamDef name="S2KTP306" orderInDisplay="0" format="NORMAL" displayIfInvalid="true"
fgColor="WHITE" bgColor="BLACK" updateWhenScroll="true"/>
</SCDdef>
```

The mapping between DBMS table and XML files is as follows:

| Field Name | SCDDef Attribute | Field Name | SCDparamDef Attribute |
|------------|------------------|------------|-----------------------|
| SPF_NUMBE  | name             | SPC_NUMBE  | -                     |
| SPF_HEAD   | description      | SPC_POS    | orderInDisplay        |
| SPF_NPAR   | -                | SPC_NAME   | name                  |
|            |                  | SPC_UPDT   | updateWhenScroll      |
|            |                  | SPC_MODE   | displayIfInvalid      |
|            |                  | SPC_FORM   | format                |
|            |                  | SPC_BACK   | bgColor               |
|            |                  | SPC_FORE   | fgColor               |



### 3.2.26. TCP Table

This table defines all telecommand packet headers.

| No. | Column   | Type   | Description                              | Restriction      |
|-----|----------|--------|--|------------------|
| 1   | TCP_ID   | vc(8)  | Name of the telecommand packet header    | Unique, not null |
| 2   | TCP_DESC | vc(24) | Textual description of the packet header | None, null       |

### 3.2.27. PCPC Table

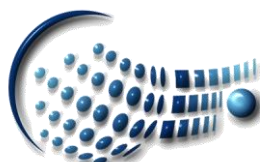
This table defines the content of all telecommand packet headers.

| No. | Column     | Type   | Description  | Restriction              |
|-----|------------|--------|--|--------------------------|
| 1   | PCPC_PNAME | vc(8)  | Telecommand packet header parameter name   | Unique, not null         |
| 2   | PCPC_DESC  | vc(24) | Textual description of the header parameter  | None, null               |
| 3   | PCPC_CODE  | ch(1)  | Flag identifying the format type in which the value is specified for this parameter (in PCDF_VALUE):<br>I = Signed integer<br>U = Unsigned integer | In set {I,U}<br>Not null |

### 3.2.28. PCDF Table

This table defines the parameter locations for all fixed size telecommand packets.

| No. | Column      | Type     | Description   | Restriction                          |
|-----|-------------|----------|---|--------------------------------------|
| 1   | PCDF_TCNAME | vc(8)    | Name of the telecommand packet header   | Valid packet header name<br>Not null |
| 2   | PCDF_DESC   | vc(24)   | Textual description of this packet header element<br><u>Note:</u><br>This field is used for fixed areas only  | None, null                           |
| 3   | PCDF_TYPE   | ch(1)    | Type of the packet header element:<br>F = Fixed area whose value is taken from PCDF_VALUE<br>A = APID parameter whose value is to be taken from CCF_APID<br>T = Service type parameter whose value is to be taken from CCF_TYPE<br>S = Service sub-type parameter whose value is to be taken from CCF_STYPE<br>K = Acknowledgement flags parameter whose value is to be taken from CCF_ACK<br>P = Other packet parameter (value is automatically set) | In set {F,A,T,S,K,P}<br>Not null     |
| 4   | PCDF_LEN    | num(4,0) | Length of the packet header element (in bits)   | 1-32<br>Not null                     |
| 5   | PCDF_BIT    | num(4,0) | Offset of the packet header element calculated from the start of the entire packet (in bits)  | 0-9999<br>Not null                   |
| 6   | PCDF_PNAME  | vc(8)    | Name of the packet header parameter<br><u>Note:</u><br>This field is only applicable if the packet header element is not of type fixed  | Valid parameter name or null         |
| 7   | PCDF_VALUE  | vc(10)   | Value of the packet header element<br><u>Note:</u><br>The value must be expressed in either decimal or with the radix specified below (for unsigned integer format types)   | Valid number<br>Not null             |



# BINARY SPACE

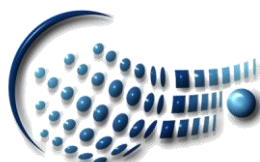
## RELIABLE SPACE SYSTEMS

|   |            |       |   |                            |
|---|------------|-------|---|----------------------------|
| 8 | PCDF_RADIX | ch(1) | Packet header element value radix:<br>D = Decimal<br>O = Octal<br>H = Hexadecimal<br><u>Note:</u><br>This field is only applicable if the parameter has an unsigned integer format type or if the packet header element is a fixed area | In set {D,O,H}<br>Not null |
|---|------------|-------|---|----------------------------|

### 3.2.29. CCF Table

This table defines all telecommands.

| No. | Column       | Type     | Description   | Restriction   |
|-----|--------------|----------|---|---|
| 1   | CCF_CNAME    | vc(8)    | Telecommand name  | Unique, not null  |
| 2   | CCF_DESCR    | vc(24)   | Textual description of the telecommand  | None, not null  |
| 3   | CCF_DESCR2   | vc(64)   | Detailed description  | None, null  |
| 4   | CCF_CTYPE    | vc(8)    | Telecommand type:<br>R = Control Segment (a.k.a. Reset Map Command)<br>F = Control Frame (a.k.a. BC frame i.e. Unlock and Set V(R))<br>S = Command Packet which does not contain the <i>Packet Error Control Field</i><br>T = SLE Throw Event<br>N = NIS Throw Event<br><u>Note:</u><br>For any other standard telecommand (i.e. except the special commands/directives listed above), this field can be left null or populated with any other character string (not matching R,F,S,T or N) with a mission specific meaning | None, null  |
| 5   | CCF_CRITICAL | ch(1)    | Criticality flag:<br>Y = Critical (hazardous)<br>N = Non-critical   | In set {Y,N}<br>Not null  |
| 6   | CCF_PKTID    | vc(8)    | Name of the packet header   | Valid packet header<br>Not null   |
| 7   | CCF_TYPE     | num(3,0) | Service type  | 0-255 or null if PCDF_TYPE=T (no entries)   |
| 8   | CCF_STYPE    | num(3,0) | Service sub-type  | 0-255 or null if PCDF_TYPE=S (no entries)   |
| 9   | CCF_APID     | num(5,0) | Application identifier  | 0-2047 for spacecraft telecommands<br>-65535 for others<br>Null if PCDF_TYPE=A (no entries) |
| 10  | CCF_NPARS    | num(3,0) | Number of elements  | 1-999 or null   |
| 11  | CCF_PLAN     | ch(1)    | Planning source flag:<br>A = All sources<br>F = Flight dynamics only<br>S = Spacecraft operations center only<br>N = Not plannable by any source  | In set {A,F,S,N}<br>Not null  |
| 12  | CCF_EXEC     | ch(1)    | Stand-alone loading flag:<br>Y = Can be loaded stand-alone<br>N = Can only be loaded as part of a telecommand sequence  | In set {Y,N}<br>Not null  |





# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |                |          |   |   |
|----|----------------|----------|---|---|
| 13 | CCF_ILSCOPE    | ch(1)    | Default interlock flag:<br>G = The telecommand will globally interlock the whole system<br>L = Interlocks only the local commanding source<br>S = Interlocks local commanding for the sub-system that the command belongs to<br>B = Interlocks global commanding only for the sub-system that the command belongs to<br>N = No associated interlock | In set {G,L,S,B,N}<br>Not null                                  |
| 14 | CCF_ILSTAGE    | ch(1)    | Default interlock verification stage type:<br>R = Reception/acceptance at the ground station<br>U = Uplink (also referred to as radiation/transmission)<br>O = On-board reception/acceptance (final uplink verification)<br>A = On-board application acceptance<br>C = Completion   | In set {R,U,O,A,C}<br>Not null                                  |
| 15 | CCF_SUBSYS     | num(3,0) | Sub-system identifier   | 1-255 or null   |
| 16 | CCF_HIPRI      | ch(1)    | High-priority flag:<br>Y = High priority telecommand<br>N = Not high priority   | In set {Y,N}<br>Not null  |
| 17 | CCF_MAPID      | num(2,0) | MAP ID value  | 0-63 if CCF_HIPRI=Y or null                                     |
| 18 | CCF_DEFSET     | vc(8)    | Default parameter value set name  | Valid parameter set name or null                                |
| 19 | CCF_RAPID      | num(5,0) | Redundant application identifier<br><u>Note:</u><br>All telecommands associated to the same prime application identifier must also be associated to the same redundant one  | 0-2047 for spacecraft telecommands<br>-65535 for others<br>Null |
| 20 | CCF_ACK        | num(2,0) | Default Acknowledgement flags   | 0-15 or null  |
| 21 | CCF_SUBSCHEDID | num(5,0) | Default on-board sub-schedule identifier  | 1-65535 or null   |

### 3.2.30. DST Table

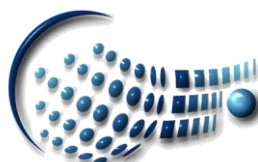
This table defines the routing path and/or the destinations for all telecommands.

| No. | Column    | Type     | Description  | Restriction                              |
|-----|-----------|----------|--|--|
| 1   | DST_APID  | num(5,0) | Telecommand application identifier   | Valid application identifier<br>Not null |
| 2   | DST_ROUTE | vc(30)   | Routing path and destination<br><u>Note:</u><br>This is mission specific information | None, not null                           |

### 3.2.31. CPC Table

This table defines all telecommand parameters.

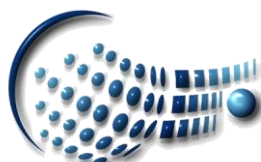
| No. | Column    | Type     | Description   | Restriction      |
|-----|-----------|----------|---|------------------|
| 1   | CPC_PNAME | vc(8)    | Telecommand parameter name  | Unique, not null |
| 2   | CPC_DESCR | vc(24)   | Textual description of the telecommand parameter  | None, null       |
| 3   | CPC_PTC   | num(2,0) | Parameter type code<br><u>Note:</u><br>This is a SCOS-2000 specific number (see Appendix A of the above mentioned document) | 1-13<br>Not null |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |             |          |  |  |
|----|-------------|----------|--|--|
| 4  | CPC_PFC     | num(5,0) | Parameter format code<br><u>Note:</u><br>This is a SCOS-2000 specific number (see Appendix A of the above mentioned document)  | 0-32<br>Not null                                     |
| 5  | CPC_DISPFMT | ch(1)    | Display format flag of the engineering values:<br>A = Text (i.e. textual calibration)<br>I = Signed integer<br>U = Unsigned integer<br>R = Floating-point<br>T = Absolute time<br>D = Delta time | In set {A,I,U,R,T,D}<br>Not null                     |
| 6  | CPC_RADIX   | ch(1)    | Default radix of the raw values:<br>D = Decimal<br>O = Octal<br>H = Hexadecimal<br><u>Note:</u><br>This field is only applicable if the parameter has an unsigned integer format type            | In set {D,O,H}<br>Not null                           |
| 7  | CPC_UNIT    | vc(4)    | Engineering unit mnemonic of the parameter values  | None, null   |
| 8  | CPC_CATEG   | ch(1)    | Calibration category of the parameter:<br>C = Numerical calibration<br>T = Textual calibration<br>A = Command ID<br>P = Parameter ID<br>N = No calibration or special processing                 | In set {C,T,A,P,N}<br>Not null                       |
| 9  | CPC_PRFREF  | vc(10)   | Parameter range set  | Valid parameter range set name or null               |
| 10 | CPC_CCAREF  | vc(10)   | Numerical calibration curve name   | Valid calibration curve name or null if CPC_CATEG<>C |
| 11 | CPC_PAFREF  | vc(10)   | Textual calibration table name   | Valid calibration table name or null if CPC_CATEG<>T |
| 12 | CPC_INTER   | ch(1)    | Default value interpretation flag:<br>R = The value is raw<br>E = The value is expressed using an engineering representation   | In set {R,E}<br>Not null                             |
| 13 | CPC_DEFVAL  | vc(17)   | Default value<br><u>Note:</u><br>The value must be expressed as specified by the fields CPC_INTER and CPC_DISPFMT/CPC_RADIX  | Valid number or null                                 |
| 14 | CPC_CORR    | ch(1)    | Time correlation flag:<br>Y = Requires time correlation<br>N = No time correlation<br><u>Note:</u><br>This field only applies to parameters representing relative or absolute times              | In set {Y,N}<br>Not null                             |
| 15 | CPC_OBTID   | num(5,0) | On-board timer identifier  | 1-99999 or null if CPC_CORR<>Y                       |



### 3.2.32. CDF Table

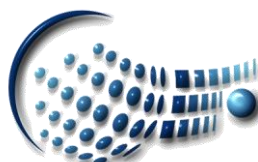
This table defines the application data field content for all telecommands.

| No. | Column      | Type     | Description   | Restriction                                     |
|-----|-------------|----------|---|---|
| 1   | CDF_CNAME   | vc(8)    | Telecommand name  | Valid telecommand name<br>Not null              |
| 2   | CDF_ELTYPE  | ch(1)    | Telecommand element type:<br>A = Fixed area<br>F = Non-editable parameter<br>E = Editable parameter   | In set {A,F,E}<br>Not null                      |
| 3   | CDF_DESCR   | vc(24)   | Textual description of the element  | Null if CDF_ELTYPE<>A                           |
| 4   | CDF_ELLEN   | num(4,0) | Element width (in bits)   | 1-32 or 0 if<br>CDF_ELTYPE<>A                   |
| 5   | CDF_BIT     | num(4,0) | Element offset (in bits)<br><u>Note:</u><br>This field specifies the bit offset of this element relative to the start of the unexpanded telecommand application data field  | 0-9999<br>Not null                              |
| 6   | CDF_GRPsize | num(2,0) | Group size (in parameters)  | 1-99 or 0                                       |
| 7   | CDF_PNAME   | vc(8)    | Telecommand parameter   | Valid parameter name or null<br>if CDF_ELTYPE=A |
| 8   | CDF_INTER   | ch(1)    | Default value interpretation flag:<br>R = The value is raw (use CDF_VALUE)<br>E = The value is expressed using an engineering representation (use CDF_VALUE)<br>D = Use the value specified in CPC_DEFVAL<br>T = Use the value provided by the telemetry parameter specified in CDF_TMID  | In set {R,E,D,T}<br>Not null                    |
| 9   | CDF_VALUE   | vc(17)   | Default or fixed value<br><u>Note:</u><br>- If the element type is fixed, this value must be expressed with a hexadecimal radix<br>- If it is a parameter, this value must be expressed as specified by CDF_INTER i.e. raw values must conform with CPC_PTC and CPC_RADIX and engineering values with CPC_DISPFMT<br>- If this element is a parameter of type <i>Command ID</i> , this field (if not null) shall contain the name of a telecommand not containing any parameter without default value<br>- If this element is a parameter of type <i>Parameter ID</i> , this field (if not null) shall contain the name of a monitoring parameter which is associated to an on-board identifier<br>- This field can take 32 bits value at maximum | Valid value or null                             |
| 10  | CDF_TMID    | vc(8)    | Monitoring parameter used to extract dynamic default values   | Valid parameter name or null<br>if CDF_INTER<>T |

### 3.2.33. CSF Table

This table defines all telecommand sequences.

| No. | Column    | Type   | Description                                     | Restriction      |
|-----|-----------|--------|---|------------------|
| 1   | CSF_NAME  | vc(8)  | Telecommand sequence name                       | Unique, not null |
| 2   | CSF_DESC  | vc(24) | Textual description of the telecommand sequence | None, not null   |
| 3   | CSF_DESC2 | vc(64) | Detailed description                            | None, null       |



# BINARY SPACE

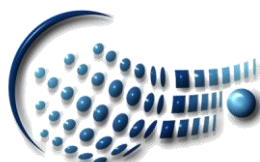
## RELIABLE SPACE SYSTEMS

|    |                |          |  |                                  |
|----|----------------|----------|--|----------------------------------|
| 4  | CSF_IFFT       | ch(1)    | Execution time-tagged commands flag:<br>Y = Contains time-tagged telecommands<br>B = User selectable (i.e. on-board execution times can be changed interactively)<br>N = No time-tagged telecommands | In set {Y,B,N}<br>Not null       |
| 5  | CSF_NFPARS     | num(3,0) | Number of formal parameters  | Valid number or null             |
| 6  | CSF_ELEMS      | num(5,0) | Number of elements in this sequence  | Valid number or null             |
| 7  | CSF_CRITICAL   | ch(1)    | Criticality flag:<br>Y = Critical (hazardous)<br>N = Non-critical  | In set {Y,N}<br>Not null         |
| 8  | CSF_PLAN       | ch(1)    | Planning source flag:<br>A = All sources<br>F = Flight dynamics only<br>S = Spacecraft operations center only<br>N = Not plannable by any source   | In set {A,F,S,N}<br>Not null     |
| 9  | CSF_EXEC       | ch(1)    | Stand-alone loading flag:<br>Y = Can be loaded stand-alone<br>N = Can only be loaded as part of a telecommand sequence   | In set {Y,N}<br>Not null         |
| 10 | CSF_SUBSYS     | num(3,0) | Sub-system identifier  | 1-255 or null                    |
| 11 | CSF_GENTIME    | vc(17)   | Time when sequence was generated   | Valid time or null               |
| 12 | CSF_DOCNAME    | vc(32)   | Document name from which it was generated  | None, null                       |
| 13 | CSF_ISSUE      | vc(10)   | Issue number of document described above   | None, null                       |
| 14 | CSF_DATE       | vc(17)   | Issue date of document described above   | None, null                       |
| 15 | CSF_DEFSET     | vc(8)    | Default parameter value set name   | Valid parameter set name or null |
| 16 | CSF_SUBSCHEDID | num(5,0) | Default on-board sub-schedule identifier   | 1-65535 or null                  |

### 3.2.34. CSS Table

This table defines all telecommand sequence elements.

| No. | Column     | Type     | Description   | Restriction                     |
|-----|------------|----------|---|---------------------------------|
| 1   | CSS_SQNAME | vc(8)    | Telecommand sequence name   | Valid sequence name<br>Not null |
| 2   | CSS_COMM   | vc(32)   | Additional comment  | None, not null if CSS_TYPE=T    |
| 3   | CSS_ENTRY  | num(5,0) | Element entry number  | 1-99999<br>Not null             |
| 4   | CSS_TYPE   | ch(1)    | Element type:<br>C = Telecommand<br>S = Telecommand sequence<br>T = Textual comment<br>F = Formal parameter representing a telecommand<br>P = Formal parameter representing a telecommand sequence  | In set {C,S,T,F,P}<br>Not null  |
| 5   | CSS_ELEMID | vc(8)    | Element identifier<br><u>Note:</u><br>- If the element is a telecommand, it must be a valid telecommand name not containing any parameter of type group repeater<br>- If it is a telecommand sequence, it must be a valid sequence name<br>- If it is a formal parameter, it must be a valid formal parameter name<br>- For a textual comment it must be null | Not null if CSS_TYPE<>T         |

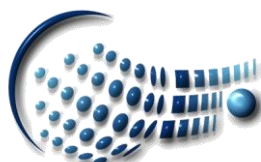




# BINARY SPACE

## RELIABLE SPACE SYSTEMS

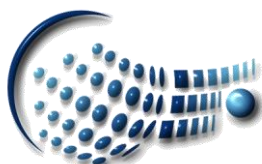
|   |             |          |  |                                   |
|---|-------------|----------|--|-----------------------------------|
| 6 | CSS_NPARS   | num(3,0) | Number of editable parameters this element has   | 0-999 or null                     |
| 7 | CSS_MANDISP | ch(1)    | Manual dispatch flag:<br>Y = The telecommand associated with this element requires a separate manual dispatch<br>N = No manual dispatch is required<br><u>Note:</u><br>This field must be left null if this element is a telecommand belonging to a group/block, unless this is the first one in the group/block   | In set {Y,N} or null              |
| 8 | CSS_RELTYPE | ch(1)    | Release delta time interpretation flag:<br>R = This element's release/start time has to be calculated relative to the release time of the previous telecommand in the expanded sequence (or relative to its own dispatch time in case this is a telecommand associated to a manual dispatch flag)<br>A = The release/start time has to be calculated relative to the start time of the sequence that this element belongs to   | In set {R,A} or null              |
| 9 | CSS_RELTIME | vc(8)    | Delta release time<br><u>Note:</u><br><ul style="list-style-type: none"> <li>- If this element is a telecommand and CSS_RELTYPE=R &amp; CSS_MANDISP=Y, then the absolute release time is calculated by adding this field to the dispatch time</li> <li>- If the same conditions as above apply but with the exception of CSS_MANDISP=N, then the absolute release time of this telecommand is calculated by adding this field to the release time of the last telecommand prior to this one in the expanded top level parent sequence</li> <li>- If this element is a telecommand and CSS_RELTYPE=A, then the absolute release time of this telecommand is calculated by adding this field to the release start time of the current sequence</li> <li>- If this element is a sequence and CSS_RELTYPE=R, then the release start time of this nested sequence is calculated by adding this field to the release time of the last telecommand in the expanded parent sequence prior to this nested sequence</li> <li>- If this element is a sequence and CSS_RELTYPE=A, then the release start time of this nested sequence is calculated by adding this field to the release start time of the current sequence</li> <li>- If this field is null and this element is a telecommand, it implies that it is not release time-tagged (ASAP release condition)</li> <li>- If this field is null and this element is a sequence, it implies that this sequence has zero delta release start time</li> <li>- This field must be left null if this element is a telecommand belonging to a group/block, unless this is the first one in the group/block</li> </ul> | Valid delta time hh:mm:ss or null |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|    |             |        |   |  |
|----|-------------|--------|---|--|
| 10 | CSS_EXTIME  | vc(17) | <p>Delta execution time</p> <p><u>Note:</u></p> <ul style="list-style-type: none"> <li>- If this element is a telecommand and CSS_PREVREL=R, then the absolute execution time is calculated by adding this field to the execution time of the last execution time-tagged telecommand prior to this one in the expanded top level parent sequence</li> <li>- If this element is a telecommand and CSS_PREVREL=A, then the absolute execution time is calculated by adding this field to the execution start time of the current sequence</li> <li>- If this element is a sequence and CSS_PREVREL=R, then the execution start time of this nested sequence is calculated by adding this field to the execution time of the last execution time-tagged telecommand prior to this one in the expanded top level parent sequence</li> <li>- If this element is a sequence and CSS_PREVREL=A, then the execution start time of this nested sequence is calculated by adding this field to the execution start time of the current sequence</li> <li>- If this field is null and this element is a telecommand, it implies that it is not execution time-tagged</li> <li>- If this field is null and this element is a sequence, it implies that this sequence has zero delta execution start time</li> </ul> | Valid delta time<br>[-]d.hhh:mm:ss or null |
| 11 | CSS_PREVREL | ch(1)  | <p>Execution delta time interpretation flag:</p> <p>R = This element's execution/start time has to be calculated relative to the execution time-tag of the previous time-tagged telecommand in the expanded sequence</p> <p>A = The execution/start time has to be calculated relative to the start time of the sequence that this element belongs to</p>   | In set {R,A} or null                       |
| 12 | CSS_GROUP   | ch(1)  | <p>Grouping condition flag:</p> <p>S = Start of the group</p> <p>M = Middle of a group</p> <p>E = End of the group</p> <p><u>Note:</u></p> <ul style="list-style-type: none"> <li>- This field is only applicable for telecommands</li> <li>- Only the first telecommand in the group is allowed to be associated to the manual dispatch flag and/or to a release time-tag</li> </ul>   | In set {S,M,E} or null                     |
| 13 | CSS_BLOCK   | ch(1)  | <p>Blocking condition flag:</p> <p>S = Start of the block</p> <p>M = Middle of a block</p> <p>E = End of the block</p> <p><u>Note:</u></p> <ul style="list-style-type: none"> <li>- This field is only applicable for telecommands</li> <li>- Only the first telecommand in the block is allowed to be associated to the manual dispatch flag and/or to a release time-tag</li> <li>- It is not allowed to include high-priority commands in a block</li> </ul>   | In set {S,M,E} or null                     |



# BINARY SPACE

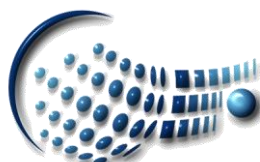
## RELIABLE SPACE SYSTEMS

|    |             |       |   |                                |
|----|-------------|-------|---|--------------------------------|
| 14 | CSS_ILSCOPE | ch(1) | <p>Interlock flag:</p> <p>G = The telecommand will globally interlock the whole system</p> <p>L = Interlocks only the local commanding source</p> <p>S = Interlocks local commanding for the sub-system that the telecommand belongs to</p> <p>B = Interlocks global commanding only for the sub-system that the telecommand belongs to</p> <p>F = Interlocks only the local commanding source and only in case of verification failure (i.e. no WAIT state)</p> <p>T = Interlocks local commanding for the sub-system that the telecommand belongs to and only in case of verification failure (i.e. no WAIT state)</p> <p>N = No associated interlock</p> <p><u>Note:</u></p> <ul style="list-style-type: none"> <li>- This field only applicable for telecommands</li> <li>- In the case of telecommands belonging to a block/group, options G and L are only allowed for the last telecommand in the block/group</li> </ul> | In set {G,L,S,B,F,T,N} or null |
| 15 | CSS_ILSTAGE | ch(1) | <p>Interlock verification stage type:</p> <p>R = Reception/acceptance at the ground station</p> <p>U = Uplink (also referred to as radiation/transmission)</p> <p>O = On-board reception/acceptance (final uplink verification)</p> <p>A = On-board application acceptance</p> <p>C = Completion</p> <p><u>Note:</u></p> <p>This field only applicable for telecommands</p>   | In set {R,U,O,A,C} or null     |
| 16 | CSS_DYNPTV  | ch(1) | <p>Dynamic pre-transmission validation check flag:</p> <p>Y = Override the related telecommand flag</p> <p>N = Use the telecommand's flag</p> <p><u>Note:</u></p> <p>This field only applicable for telecommands</p>  | In set {Y,N} or null           |
| 17 | CSS_STAPTV  | ch(1) | <p>Pre-transmission validation check flag:</p> <p>Y = Override the related telecommand flag</p> <p>N = Use the telecommand's flag</p> <p><u>Note:</u></p> <p>This field only applicable for telecommands</p>  | In set {Y,N} or null           |
| 18 | CSS_CEV     | ch(1) | <p>Telecommand execution verification flag:</p> <p>Y = Override the related telecommand flag</p> <p>N = Use the telecommand's flag</p> <p><u>Note:</u></p> <p>This field only applicable for telecommands</p>   | In set {Y,N} or null           |

### 3.2.35. SDF Table

This table defines all telecommand sequence element parameters.

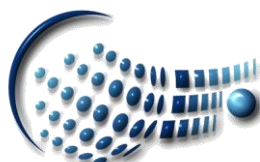
| No. | Column     | Type     | Description               | Restriction                     |
|-----|------------|----------|---------------------------|---------------------------------|
| 1   | SDF_SQNAME | vc(8)    | Telecommand sequence name | Valid sequence name<br>Not null |
| 2   | SDF_ENTRY  | num(5,0) | Element entry number      | 1-99999<br>Not null             |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|   |            |          |   |  |
|---|------------|----------|---|--|
| 3 | SDF_ELEMID | vc(8)    | Element name  | Valid telecommand, sequence or formal parameter name<br>Not null |
| 4 | SDF_POS    | num(4,0) | Element position<br><u>Note:</u><br>- If the sequence element is a telecommand, then this field specifies the bit offset of the parameter in the command<br>If telecommands containing group repeaters are used, the value of this field shall be the bit offset expected for the current parameter following expansion of the groups<br>- If it is a nested sequence, then this field specifies the sequence parameter position  | 0-9999<br>Not null   |
| 5 | SDF_PNAME  | vc(8)    | Element parameter name  | Valid (formal) parameter name<br>Not null                        |
| 6 | SDF_FTYPE  | ch(1)    | Element value modification flag:<br>F = Non-modifiable value<br>E = Editable value  | In set {F,E}<br>Not null   |
| 7 | SDF_VTYPE  | ch(1)    | Element value source flag:<br>R = Use the raw value specified in SDF_VALUE<br>E = Use the calibrated value specified in SDF_VALUE<br>F = Use the value attached to the formal parameter specified in SDF_VALUE<br>P = Use the value specified by the parameter value set in SDF_VALSET<br>S = Use the value in CSP_DEFVAL attached to the formal parameter specified by this element<br>D = Use the value specified at the corresponding position of the telecommand (CDF_VALUE)  | In set {R,E,F,P,S,D}<br>Not null                                 |
| 8 | SDF_VALUE  | vc(17)   | Element value<br><u>Note:</u><br>- If SDF_VTYPE is R or E, this value must be expressed using the appropriate representation i.e. raw values must be expressed in a format compatible with the type of the parameter, engineering values must be expressed in a format compatible with the engineering representation of the parameter<br>- If SDF_VTYPE is R and this element parameter is of type <i>Command ID</i> , this field shall contain the name of a telecommand not containing any parameter without default value; if this element parameter is of type <i>Parameter ID</i> , this field shall contain the name of a monitoring parameter which is associated to an on-board identifier<br>- If SDF_VTYPE is F, this field contains the name of the formal parameter which this element parameter will inherit the value from | Valid value or null if SDF_VTYPE in set {D,P,S}                  |
| 9 | SDF_VALSET | vc(8)    | Parameter value set value name  | Valid parameter set name or null if SDF_VTYPE<>P                 |





|    |             |          |  |                |
|----|-------------|----------|--|----------------|
| 10 | SDF_REPPPOS | num(4,0) | Telecommand parameter repeated position<br><u>Note:</u><br>This field specifies the original bit offset of the parameter in the telecommand definition | 0-9999 or null |
|----|-------------|----------|--|----------------|

### 3.2.36. CSP Table

This table defines all telecommand sequence parameters.

| No. | Column      | Type     | Description   | Restriction                              |
|-----|-------------|----------|---|--|
| 1   | CSP_SQNAME  | vc(8)    | Telecommand sequence name   | Valid sequence name<br>Not null          |
| 2   | CSP_FPNAME  | vc(8)    | Formal parameter name   | Unique (inside the sequence)<br>Not null |
| 3   | CSP_FPNUM   | num(5,0) | Position number   | 0-65535<br>Not null                      |
| 4   | CSP_DESCR   | vc(24)   | Textual description of the formal parameter   | None, null                               |
| 5   | CSP_PTC     | num(2,0) | Parameter type code<br><u>Note:</u><br>This is a SCOS-2000 specific number (see Appendix A of the above mentioned document)   | 1-13<br>Not null                         |
| 6   | CSP_PFC     | num(5,0) | Parameter format code<br><u>Note:</u><br>This is a SCOS-2000 specific number (see Appendix A of the above mentioned document)   | 0-32<br>Not null                         |
| 7   | CSP_DISPFMT | ch(1)    | Display format flag of the engineering values:<br>A = Text (i.e. textual calibration)<br>I = Signed integer<br>U = Unsigned integer<br>R = Floating-point<br>T = Absolute time<br>D = Delta time  | In set {A,I,U,R,T,D}<br>Not null         |
| 8   | CSP_RADIX   | ch(1)    | Default radix of the raw values:<br>D=Decimal radix<br>O=Octal radix<br>H=Hexadecimal radix<br><u>Note:</u><br>This field is only applicable if the parameter has an unsigned integer format type | In set {D,O,H}<br>Not null               |
| 9   | CSP_TYPE    | ch(1)    | Formal parameter type:<br>C=Represents a telecommand<br>S=Represents a telecommand sequence<br>P=Parameter associated to one or more element parameters   | In set {C,S,P}<br>Not null               |
| 10  | CSP_VTYPE   | ch(1)    | Value interpretation flag:<br>R = The value is raw<br>E = The value is expressed using an engineering representation  | In set {R,E} or null                     |



# BINARY SPACE

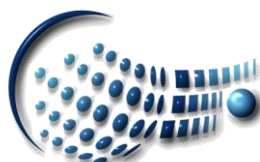
## RELIABLE SPACE SYSTEMS

|    |            |        |  |  |
|----|------------|--------|--|--|
| 11 | CSP_DEFVAL | vc(17) | Parameter value<br><u>Note:</u><br>- For 'true' formal parameters this value must be expressed using the appropriate representation as specified in CSP_VTYPE i.e. raw values must be expressed in a format compatible with the type of the parameter, engineering values must be expressed in a format compatible with the engineering representation of the parameter<br>If it is of type <i>Command ID</i> , this field (if not null) shall contain the name of a telecommand not containing any parameter without default value;<br>if it is of type <i>Parameter ID</i> , this field (if not null) shall contain the name of a monitoring parameter which is associated to an on-board identifier<br>- If this formal parameter represents a telecommand, this field must contain the name of a valid telecommand<br>- If this formal parameter represents a sequence, this field must contain the name of a valid sequence | Valid value or null                                  |
| 12 | CSP_CATEG  | ch(1)  | Calibration category of the parameter:<br>C = Numerical calibration<br>T = Textual calibration<br>A = Command ID<br>P = Parameter ID<br>N = No calibration or special processing   | In set {C,T,A,P,N}<br>Not null                       |
| 13 | CSP_PRFREF | vc(10) | Parameter range set name   | Valid parameter range set name or null               |
| 14 | CSP_CCAREF | vc(10) | Numerical calibration curve name   | Valid calibration curve name or null if CSP_CATEG<>C |
| 15 | CSP_PAFREF | vc(10) | Textual calibration table name   | Valid calibration table name or null if CSP_CATEG<>T |
| 16 | CSP_UNIT   | vc(4)  | Engineering unit mnemonic of the parameter values  | None, null   |

### 3.2.37. PTV Table

This table defines all telecommand pre-transmission validation checks.

| No. | Column     | Type  | Description  | Restriction                        |
|-----|------------|-------|--|------------------------------------|
| 1   | PTV_CNAME  | vc(8) | Telecommand name   | Valid telecommand name<br>Not null |
| 2   | PTV_PARNAM | vc(8) | Telemetry parameter name<br><u>Note:</u><br>Only monitoring parameters whose raw representation is of numerical nature (i.e. no strings, times) can be specified | Valid parameter name<br>Not null   |
| 3   | PTV_INTER  | ch(1) | Value interpretation flag:<br>R = The value is raw<br>E = The value is expressed using an engineering representation   | In set {R,E}<br>Not null           |



|   |         |        |  |                          |
|---|---------|--------|--|--------------------------|
| 4 | PTV_VAL | vc(17) | Check value<br><u>Note:</u><br>This value must be expressed as specified by PTV_INTER i.e. raw values must conform with PCF_PTC (decimal radix) and engineering values with PCF_CATEG and CAF_ENGFMT (for numerically calibrated parameters) | Valid number<br>Not null |
|---|---------|--------|--|--------------------------|

### 3.2.38. CVS Table

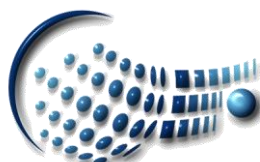
This table defines all telecommand verification stages.

| No. | Column          | Type      | Description  | Restriction                    |
|-----|-----------------|-----------|--|--------------------------------|
| 1   | CVS_ID          | num(5,0)  | Verification stage identification number   | 0-32767<br>Not null            |
| 2   | CVS_TYPE        | ch(1)     | Associated execution profile:<br>A =On-board application acceptance<br>S =Execution start<br>0-9 =Execution progress number<br>C =Completion | In set {A,S,0-9,C}<br>Not null |
| 3   | CVS_SOURCE      | ch(1)     | Verification check type flag:<br>R = Checks are based on telemetry report packets<br>V = Parameter values are used for the checks            | In set {R,V}<br>Not null       |
| 4   | CVS_START       | num(5,0)  | Delta time after release/execution time (in seconds)   | 0-99999<br>Not null            |
| 5   | CVS_INTERVAL    | num(5,0)  | Length of the sampling interval (in seconds)   | 1-99999<br>Not null            |
| 6   | CVS_SPID        | num(10,0) | Close window telemetry packet  | Valid packet name or null      |
| 7   | CVS_UNCERTAINTY | num(5,0)  | Uncertainty window size (in seconds)   | 0-99999 or -1<br>Not null      |

### 3.2.39. CVE Table

This table defines the telemetry parameters belonging to the various telecommand verification stages.

| No. | Column     | Type     | Description  | Restriction  |
|-----|------------|----------|--|--|
| 1   | CVE_CVSID  | num(5,0) | Verification stage identification number   | Valid verification stage identification number<br>Not null |
| 2   | CVE_PARNAM | vc(8)    | Telemetry parameter name<br><u>Note:</u><br>- Only monitoring parameters whose raw representation is of numerical nature (i.e. no strings, times) can be specified<br>- In case this telemetry parameter is subject to status consistency checking, then the representation used to specify the verification value shall match with the representation used to specify the out-of-limit checks | Valid parameter name<br>Not null                           |

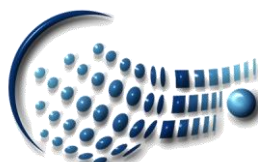


|   |           |        |   |                            |
|---|-----------|--------|---|----------------------------|
| 3 | CVE_INTER | ch(1)  | Value interpretation flag:<br>R = The value and tolerance is raw<br>E = Both values are expressed using an engineering representation<br>C = The value is a telecommand parameter name and the tolerance is expressed using a raw representation (consistency checks)<br><u>Note:</u><br>In case the same monitoring parameter is used in several parameter-based verification stages, the engineering representation can only be used if all instances consistently use it   | In set {R,E,C}<br>Not null |
| 4 | CVE_VAL   | vc(17) | Check value<br><u>Note:</u><br>- This value must be expressed using the appropriate representation i.e. raw unless option CVE_INTER=E is used and raw values must be expressed in decimal in a format compatible with the type of the verification parameter; engineering values must be expressed in a format compatible with the engineering representation of the verification parameter<br>- In case of echo verification i.e. CVE_INTER=C, this field contains the name of the telecommand parameter whose value at telecommand instantiation is used to check against to perform the verification check | Valid value or null        |
| 5 | CVE_TOL   | vc(17) | Check value tolerance<br><u>Note:</u><br>- This value must be expressed using the same representation as CVE_VAL<br>- It must be null for calibrated status telemetry parameters  | Valid number or null       |
| 6 | CVE_CHECK | ch(1)  | Check choice flag:<br>B = This check has to be used both for actual verification and to disable/re-enable status consistency checking<br>S = Disable status consistency checking  | In set {B,S}<br>Not null   |

### 3.2.40. CVP Table

This table defines all associations between telecommand verification stages and telecommands (or telecommand sequences).

| No. | Column    | Type     | Description  | Restriction  |
|-----|-----------|----------|--|--|
| 1   | CVP_TASK  | vc(8)    | Task name  | Valid telecommand (or sequence) name<br>Not null           |
| 2   | CVP_TYPE  | ch(1)    | Type flag:<br>C = The specified task is a telecommand<br>S = The task name represents a telecommand sequence | In set {C,S}<br>Not null                                   |
| 3   | CVP_CVSID | num(5,0) | Verification stage identification number   | Valid verification stage identification number<br>Not null |





### 3.2.41. PST Table

This table defines all telecommand parameter sets.

| No. | Column    | Type   | Description                              | Restriction      |
|-----|-----------|--------|--|------------------|
| 1   | PST_NAME  | vc(8)  | Telecommand parameter set name           | Unique, not null |
| 2   | PST_DESCR | vc(24) | Textual description of the parameter set | None, null       |

### 3.2.42. PSV Table

This table defines all telecommand parameter value sets (including the associations to the telecommand parameter sets).

| No. | Column    | Type   | Description                                    | Restriction                          |
|-----|-----------|--------|--|--------------------------------------|
| 1   | PSV_NAME  | vc(8)  | Telecommand parameter set name                 | Valid parameter set name<br>Not null |
| 2   | PSV_PVSID | vc(8)  | Telecommand parameter value set name           | Unique, not null                     |
| 3   | PSV_DESCR | vc(24) | Textual description of the parameter value set | None, null                           |

### 3.2.43. CPS Table

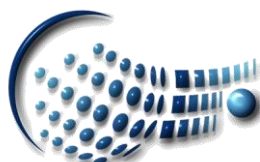
This table defines the content of all telecommand parameter sets.

| No. | Column   | Type     | Description   | Restriction                               |
|-----|----------|----------|---|---|
| 1   | CPS_NAME | vc(8)    | Telecommand parameter set name  | Valid parameter set name<br>Not null      |
| 2   | CPS_PAR  | vc(8)    | Telecommand parameter name  | Valid (formal) parameter name<br>Not null |
| 3   | CPS_BIT  | num(4,0) | Bit offset or position number<br><u>Note:</u><br>- For telecommand parameters, this field contains its bit offset<br>- For formal parameters, this field contains its position number | 0-9999<br>Not null                        |

### 3.2.44. PVS Table

This table defines the content of all telecommand parameter value sets.

| No. | Column    | Type  | Description  | Restriction                                |
|-----|-----------|-------|--|--|
| 1   | PVS_ID    | vc(8) | Telecommand parameter value set name   | Valid parameter value set name<br>Not null |
| 2   | PVS_PSID  | vc(8) | Telecommand parameter set name   | Valid parameter set name<br>Not null       |
| 3   | PVS_PNAME | vc(8) | Telecommand parameter name   | Valid (formal) parameter name<br>Not null  |
| 4   | PVS_INTER | ch(1) | Value interpretation flag:<br>R = The value is raw<br>E = The value is expressed using an engineering representation | In set {R,E}<br>Not null                   |



|   |          |          |  |                     |
|---|----------|----------|--|---------------------|
| 5 | PVS_VALS | vc(17)   | Parameter value<br><u>Note:</u><br>This value must be expressed using the appropriate representation i.e. raw unless option PVS_INTER=E is used and raw values must be expressed in a format compatible with the type of the parameter; engineering values must be expressed in a format compatible with the engineering representation of the parameter | Valid value or null |
| 6 | PVS_BIT  | num(4,0) | Bit offset or position number<br><u>Note:</u><br>- For telecommand parameters, this field contains its bit offset<br>- For formal parameters, this field contains its position number  | 0-9999<br>Not null  |

### 3.2.45. PSM Table

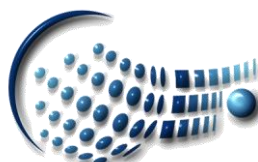
This table defines all associations between telecommand parameter sets and telecommands (or telecommand sequences).

| No. | Column     | Type  | Description  | Restriction                                      |
|-----|------------|-------|--|--|
| 1   | PSM_NAME   | vc(8) | Task name  | Valid telecommand (or sequence) name<br>Not null |
| 2   | PSM_TYPE   | ch(1) | Type flag:<br>C = The specified task is a telecommand<br>S = The task name represents a telecommand sequence | In set {C,S}<br>Not null                         |
| 3   | PSM_PARSET | vc(8) | Telecommand parameter set name   | Valid parameter set name<br>Not null             |

### 3.2.46. CCA Table

This table defines all numerical telecommand calibration curves.

| No. | Column     | Type     | Description  | Restriction                       |
|-----|------------|----------|--|-----------------------------------|
| 1   | CCA_NUMBR  | vc(10)   | Numerical calibration curve name   | Unique, not null                  |
| 2   | CCA_DESCR  | vc(24)   | Description of the calibration curve   | None, null                        |
| 3   | CCA_ENGFMT | ch(1)    | Engineering format type flag:<br>I =Signed integer<br>U=Unsigned integer<br>R=Floating-point | In set {I,U,R}<br>Not null        |
| 4   | CCA_RAWFMT | ch(1)    | Raw format type flag:<br>I =Signed integer<br>U=Unsigned integer<br>R=Floating-point         | In set {I,U,R}<br>Not null        |
| 5   | CCA_RADIX  | ch(1)    | Radix for the raw format type:<br>D =Decimal<br>H =Hexadecimal<br>O=Octal                    | In set {D,H,O} if<br>CCA_RAWFMT=U |
| 6   | CCA_UNIT   | vc(4)    | Engineering unit mnemonic  | None, null                        |
| 7   | CCA_NCURVE | num(3,0) | Number of points defined for this calibration curve  | 0-999<br>Not null                 |



### 3.2.47. CCS Table

This table defines all points allocated to numerical telecommand calibration curves.

| No. | Column    | Type   | Description  | Restriction                              |
|-----|-----------|--------|--|--|
| 1   | CCS_NUMBR | vc(10) | Numerical calibration curve name   | Valid calibration curve name<br>Not null |
| 2   | CCS_XVALS | vc(17) | Raw value of the calibration point<br><u>Note:</u><br>This value has to be expressed in a format compatible with CCA_RAWFMT; unsigned integer values (CCA_RAWFMT=U) have to be expressed using the radix associated to this calibration curve (CCA_RADIX)                                  | Valid raw number<br>Not null             |
| 3   | CCS_YVALS | vc(17) | Engineering value of the calibration point<br><u>Note:</u><br>This value has to be expressed in a format compatible with 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) | Valid calibrated value<br>Not null       |

### 3.2.48. PAF Table

This table defines all textual telecommand calibration curves.

| No. | Column     | Type     | Description   | Restriction                |
|-----|------------|----------|---|----------------------------|
| 1   | PAF_NUMBR  | vc(10)   | Textual calibration table name  | Unique, not null           |
| 2   | PAF_DESCR  | vc(24)   | Description of the calibration table  | None, null                 |
| 3   | PAF_RAWFMT | ch(1)    | Raw format type flag:<br>I = Signed integer<br>U = Unsigned integer<br>R = Floating-point | In set {I,U,R}<br>Not null |
| 4   | PAF_NALIAS | num(3,0) | Number of entries defined for this calibration table                                      | 0-999<br>Not null          |

### 3.2.49. PAS Table

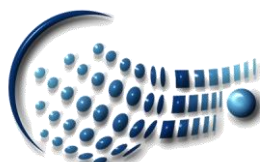
This table defines all entries allocated to textual telecommand calibration tables.

| No. | Column    | Type   | Description   | Restriction                              |
|-----|-----------|--------|---|--|
| 1   | PAS_NUMBR | vc(10) | Textual calibration table name  | Valid calibration table name<br>Not null |
| 2   | PAS_ALTXT | vc(16) | Engineering value text  | None, not null                           |
| 3   | PAS_ALVAL | vc(17) | Raw value<br><u>Note:</u><br>This value has to be expressed (in decimal) in a format compatible with PAF_RAWFMT | Valid number<br>Not null                 |

### 3.2.50. PRF Table

This table defines the numerical/textual telecommand out-of-limit check tables.

| No. | Column    | Type   | Description  | Restriction              |
|-----|-----------|--------|--|--------------------------|
| 1   | PRF_NUMBR | vc(10) | Telecommand parameter range set name                                     | Unique, not null         |
| 2   | PRF_DESCR | vc(24) | Textual description of the parameter range set                           | None, null               |
| 3   | PRF_INTER | ch(1)  | Limit values interpretation flag:<br>R=Raw values<br>E=Calibrated values | In set {R,E}<br>Not null |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|   |            |          |  |                                  |
|---|------------|----------|--|----------------------------------|
| 4 | PRF_DSPFMT | ch(1)    | Display format flag of the engineering values:<br>A = Text (i.e. textual calibration)<br>I = Signed integer<br>U = Unsigned integer<br>R = Floating-point<br>T = Absolute time<br>D = Delta time | In set {A,I,U,R,T,D}<br>Not null |
| 5 | PRF_RADIX  | ch(1)    | Radix of the limit values:<br>D = Decimal<br>O = Octal<br>H = Hexadecimal<br><u>Note:</u><br>This field is only applicable if the parameter has an unsigned integer format type                  | In set {D,O,H}<br>Not null       |
| 6 | PRF_NRANGE | num(3,0) | Number of associated value limits  | 0-999 or null                    |
| 7 | PRF_UNIT   | vc(4)    | Engineering unit mnemonic of the limit values  | None, null                       |

### 3.2.51. PRV Table

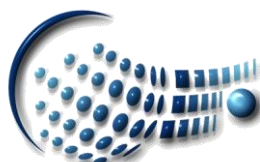
This table defines all value limit checks associated to the telecommand out-of-limit tables.

| No. | Column     | Type   | Description  | Restriction                                |
|-----|------------|--------|--|--|
| 1   | PRV_NUMBR  | vc(10) | Telecommand parameter range set name   | Valid parameter range set name<br>Not null |
| 2   | PRV_MINVAL | vc(17) | Low limit value<br><u>Note:</u><br>This value must be expressed using the appropriate representation (see PRF_INTER); raw values must be expressed in a format compatible with the type of the parameter; engineering values must be expressed in a format compatible with the engineering representation specified by PRF_DSPFMT/PRF_RADIX (for unsigned integer display format only) | Valid limit value<br>Not null              |
| 3   | PRV_MAXVAL | vc(17) | High limit value<br><u>Note:</u><br>- This field can be left null indicating that the low value is a discrete value<br>- If specified, the value must be expressed using the same representation as the low limit value  | Valid limit value or null                  |

### 3.2.52. GEN Table

This table provides the complete list of on-board processors.

| No. | Column        | Type   | Description                                      | Restriction                            |
|-----|---------------|--------|--|--|
| 1   | GEN_PROCESSOR | vc(10) | On-board processor name                          | Unique, not null                       |
| 2   | GEN_PATCHTC   | vc(12) | Telecommand function used to patch the processor | Valid telecommand function or null     |
| 3   | GEN_DUMP TC   | vc(12) | Telecommand function used to dump the processor  | Valid telecommand function<br>Not null |
| 4   | GEN_TPKT      | vc(4)  | Telemetry packet in which the dump arrives       | Valid telemetry packet<br>Not null     |
| 5   | GEN_STARTPAGE | int    | Page number where the processor memory starts    | 0-2 <sup>32</sup> -1<br>Not null       |
| 6   | GEN_STARTADDR | int    | Address where the processor memory starts        | 0-2 <sup>32</sup> -1<br>Not null       |





|    |              |       |  |                                  |
|----|--------------|-------|--|----------------------------------|
| 7  | GEN_ENDPAGE  | int   | Page number where the processor memory ends        | 0-2 <sup>32</sup> -1<br>Not null |
| 8  | GEN_ENDADDR  | int   | Address where the processor memory ends            | 0-2 <sup>32</sup> -1<br>Not null |
| 9  | GEN_IMGNAME  | vc(9) | Name of the master image for this processor        | Valid memory image<br>Not null   |
| 10 | GEN_INTERVAL | int   | Interval in seconds between dump packets           | >0<br>Not null                   |
| 11 | GEN_NUMPATCH | int   | Number of patch commands between each dump command | >=0 or null                      |

### 3.2.53. MAD Table

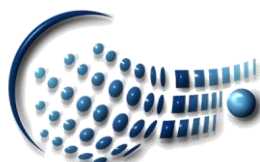
This table provides the complete list of all memory areas contained in the on-board processors.

| No. | Column        | Type   | Description  | Restriction                           |
|-----|---------------|--------|--|---------------------------------------|
| 1   | MAD_PROCESSOR | vc(10) | Name of the on-board processor                                 | Valid on-board processor<br>Not null  |
| 2   | MAD_NAME      | vc(48) | Memory area name   | Unique, not null                      |
| 3   | MAD_DESCR     | vc(64) | Description of the memory area                                 | None or null                          |
| 4   | MAD_TYPE      | vc(4)  | Memory area type:<br>MEM=Memory block<br>IMG =Memory image     | In set {MEM,IMG}<br>Not null          |
| 5   | MAD_PATCHTC   | vc(12) | Telecommand function used to patch the memory area             | Valid telecommand function<br>or null |
| 6   | MAD_DUMPTC    | vc(12) | Telecommand function used to dump the memory area              | Valid telecommand function<br>or null |
| 7   | MAD_TPKT      | vc(4)  | Telemetry packet in which the dump arrives                     | Valid telemetry packet or null        |
| 8   | MAD_PATCHABLE | ch(1)  | Flag indicating if the memory area can be patched              | In set {Y,N}<br>Not null              |
| 9   | MAD_DUMPABLE  | ch(1)  | Flag indicating if the memory area can be dumped               | In set {Y,N}<br>Not null              |
| 10  | MAD_UPDATE    | ch(1)  | Flag indicating if the ground image can be dynamically updated | In set {Y,N}<br>Not null              |
| 11  | MAD_INHIBIT   | ch(1)  | Flag indicating if the dynamic update can be inhibited         | In set {Y,N}<br>Not null              |

### 3.2.54. MEM Table

This table provides the complete list of all memory blocks contained in the memory areas.

| No. | Column      | Type   | Description  | Restriction                            |
|-----|-------------|--------|--|--|
| 1   | MEM_MNAME   | vc(9)  | Name of the memory area  | Valid memory area<br>Not null          |
| 2   | MEM_TMNAME  | vc(6)  | Telemetry parameter that represents the memory block           | Valid telemetry parameter or<br>null   |
| 3   | MEM_TCNAME  | vc(9)  | Telecommand parameter that is associated with the memory block | Valid telecommand<br>parameter or null |
| 4   | MEM_PAGE    | int    | Memory page this block resides in                              | 0-2 <sup>32</sup> -1<br>Not null       |
| 5   | MEM_ADDRESS | int    | Memory address this block resides in                           | 0-2 <sup>32</sup> -1<br>Not null       |
| 6   | MEM_BITOFF  | int    | Bit offset of the data   | 0-2 <sup>32</sup> -1<br>Not null       |
| 7   | MEM_WIDTH   | int    | Width (in bits) of the data                                    | 1-32 or null                           |
| 8   | MEM_DESCR   | vc(72) | Description of the memory area block                           | None or null                           |



### 3.2.55. IMG Table

This table provides the complete list of all memory images contained in the memory areas.

| No. | Column      | Type   | Description                          | Restriction                              |
|-----|-------------|--------|--------------------------------------|--|
| 1   | IMG_MNAME   | vc(9)  | Name of the memory image             | None, not null                           |
| 2   | IMG_VERSION | vc(10) | Version of the memory image          | None<br>Not null                         |
| 3   | IMG_PAGE    | int    | Memory page this image resides in    | 0-2 <sup>32</sup> -1<br>Not null         |
| 4   | IMG_ADDRESS | int    | Memory address this image resides in | 0-2 <sup>32</sup> -1<br>Not null         |
| 5   | IMG_DATA    | text   | Data                                 | Maximal 2000 hexadecimal numbers or null |

### 3.2.56. CRC Table

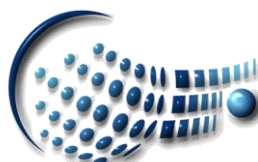
This table provides the complete list of all check sums attached to the memory areas.

| No. | Column        | Type   | Description  | Restriction                          |
|-----|---------------|--------|--|--------------------------------------|
| 1   | CRC_NAME      | vc(12) | Name of the memory area this check sum applies to                    | Valid memory area<br>Not null        |
| 2   | CRC_PROCESSOR | vc(10) | Name of the on-board processor that contains the related memory area | Valid on-board processor<br>Not null |
| 3   | CRC_DEST      | int    | The destination of the check sum                                     | 0-2 <sup>32</sup> -1 or null         |
| 4   | CRC_LENGTH    | int    | Length of the check sum  | 16, 32 or null                       |
| 5   | CRC_PAGE      | int    | Memory page the check sum applies to                                 | 0-2 <sup>32</sup> -1 or null         |
| 6   | CRC_START     | int    | Starting memory address the check sum applies to                     | 0-2 <sup>32</sup> -1 or null         |
| 7   | CRC_END       | int    | Ending memory address the check sum applies to                       | 0-2 <sup>32</sup> -1 or null         |
| 8   | CRC_SEED      | vc(8)  | Seed value of the check sum  | Value (hex) or null                  |

### 3.2.57. CAT Table

This table provides the complete list of all catalogued memory area versions.

| No. | Column       | Type   | Description  | Restriction                   |
|-----|--------------|--------|--|-------------------------------|
| 1   | CAT_ID       | vc(12) | Catalogue name                                     | None, not null                |
| 2   | CAT_NAME     | vc(9)  | Name of the memory area                            | Valid memory area<br>Not null |
| 3   | CAT_DESCR    | vc(32) | Description of the catalogue entry                 | None or null                  |
| 4   | CAT_VERSION  | vc(10) | Version of the memory area catalogued              | None<br>Not null              |
| 5   | CAT_DATE     | vc(32) | Date the memory area was catalogued                | None or null                  |
| 6   | CAT_TIME     | vc(32) | Time the memory area was catalogued                | None or null                  |
| 7   | CAT_LOCATION | vc(12) | Storage location of the memory area                | None or null                  |
| 8   | CAT_COMPLETE | ch(1)  | Flag indicating if the catalogue entry is complete | In set {Y,N} or null          |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

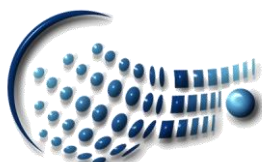
Satview™ supports a custom on-board system memory handling database definition. It, however, allows the import/export of SCOS-2000 compatible memory model and memory image files (see the document 'SCOS-2000 OBSM External Interfaces ICD').

The following table describes the structure of the memory model import/export file header:

| Line | Name                   | Size | Description   | Restriction                |
|------|------------------------|------|---|----------------------------|
| 1    | Description            | 64   | Free textual description of the memory model used for display purposes  | None, not null             |
| 2    | Category               | 1    | Memory model category flag:<br>C = Configuration tables i.e. memory models which can be used for the generation of patch/dump command data and thus are supposed to contain non-overlapping (but not necessarily contiguous) memory area definitions<br>S = Symbolic tables i.e. memory models which cannot be used for the generation of command data. Typically, symbolic tables are used to provide a detailed interpretation of a memory image. They are allowed to contain 'nested' memory area definitions (i.e. areas which overlap but such that one area is completely contained in another one) | In set {C,S}<br>Not null   |
| 3    | Memory Device          | 8    | Memory device name  | None, null                 |
| 4    | Base                   | 1    | Memory address base:<br>1 =Bytes<br>2 =Words<br>4 =Double-words   | In set {1,2,4}<br>Not null |
| 5    | Domain                 | 5    | Memory model domain identifier  | 0-99999 or null            |
| 6    | Start address          | 8    | Start address of the memory model (in hex)  | Valid address or null      |
| 7    | End address            | 8    | End address of the memory model (in hex)  | Valid address or null      |
| 8    | Number of memory areas | 1    | Number of memory areas defined  | Valid number or null       |

The structure of the memory areas following the memory model header is as follows:

| No. | Name          | Size | Description   | Restriction                         |
|-----|---------------|------|---|-------------------------------------|
| 1   | Name          | 48   | Name of the memory area (also referred to as symbolic address)  | Unique, not null                    |
| 2   | Start address | 8    | Start address of the memory area (in hex)   | Valid number<br>Not null            |
| 3   | Length        | 8    | The length of the memory area (in hex)  | Valid number<br>Not null            |
| 4   | Type          | 1    | Data type:<br>V = Variable<br>D = Data<br>O = Object code<br>P = Parameters   | In set {V,D,O,P} or null (not used) |
| 5   | Patch         | 1    | Patch category flag:<br>P = Patchable i.e. the area has to be used for off-line comparison and patched only if changed<br>N = Not patchable<br>M = Mandatory i.e. the area has to be patched anyhow (even if not changed) | In set {P,N,M}<br>Not null          |



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

|   |         |    |   |                            |
|---|---------|----|---|----------------------------|
| 6 | Dump    | 1  | Dump/monitor category flag:<br>D = Dumpable only i.e. the area shall be dumped, used for off-line comparison but not used for monitoring<br>N = Not dumpable<br>M = Mandatory i.e. the area shall be dumped and shall be used for both monitoring and off-line comparison | In set {D,N,M}<br>Not null |
| 7 | Comment | 64 | Textual comment   | None, null                 |

### Notes:

- Use a separate file for each memory model definition
- Specify one line for each memory area definition
- Use CR for line separators, TABs for column delimiters

Memory images definition import/export files have the following structure:

**DEVICE**=*Memory Device Name*

**VERSION**=*Version/Issue*

**UNIT**=*Address Base*

**START**=*Start Address (hex)*,**COUNT**=*Length (hex)*,**DATA**=*Data (hex)*

{**START**=*Start Address (hex)*,**COUNT**=*Length (hex)*,**DATA**=*Data (hex)*}

### Notes:

- Use a separate file for each memory image definition
- Use CR for line separators

