



SPIRE Technical Note

Contents of a SPIRE VM Table File

S.D. Sidher

Ref: SPIRE-RAL-NOT-001907

Issue: 1.0 Draft 2

Date: 5/2/2004

Page: 1 of 3

1. INTRODUCTION

In order to support SPIRE Command Lists in the HCSS it is necessary for the CUS to be able to read the contents of SPIRE Virtual Machine (VM) table files. This note defines the structure and contents of such a VM table file. Each VM table file will be generated by the IFSI provided VM compiler and then read by the HCSS CUS before commands are uplinked to the SPIRE instrument. The contents of a VM table are essentially the input parameters to specific SPIRE instrument telecommands.

This initial version is applicable to SPIRE testing but will be updated to cover later phases.

2. REFERENCE DOCUMENTS

- 1) "CUS Functionality to Support Command Lists for SPIRE", HCSS SCR-0110 (see http://astro.esa.int/herschel_webapps/servletsuite/ProblemReportServlet?area=hcss&mode=displaypr&id=110), Ken King and Sunil Sidher, 13/08/02.
- 2) "Comparison of instrument commanding on Herschel", SPIRE-RAL-NOT-001618, Ken King and Sunil Sidher, Issue 1.1, 06/06/03.
- 3) "The SPIRE instrument operating philosophy: An HCSS analysis", FSCDT-TN037, Issue 0.2, 20/11/03.

3. VM TABLE FILE NAME

The VM Table generated by VM compiler will be written to an ASCII file with the following naming convention:

`<Command List Name>_<VM Compiler Version>_<Command List Version>_<YYMMDDHHMMSS>.tbl`

where YYMMDDHHMMSS is the date and time when the VM table is generated by the VM compiler.

Example File Name: CHOP_1.2_2.1_24678_031201123456.tbl

4. VM TABLE FILE HEADER RECORDS

The VM Table file will contain the following header records:

Record	Contents	Example
1	#CLName <CL Name>	#CLname: CHOP
2	#VMVersion <VM Version>	#VMVersion: 1.2
3	#CLVersion <CL Version>	#CLVersion: 2.1
4	#CLCVSId <CL CVS Id>	#CLCVSId: 24678
5	#TableId <Table Id>	#TableId: 0x50
6	#TableLength <Table Length>	#TableLength: 0x64
7	#Date <YYMMDDHHMMSS>	#Date: 031201123456

- Each of these records will commence with a '#' symbol in column 1.



- The TableId and TableLength will be written in either decimal or hexadecimal notation. Hex numbers will be written with a preceding '0x'.
- The TableLength represents the total number of 32 bit hexadecimal words in the table (see section 5).

5. VM TABLE FILE DATA RECORDS

The VM Table file will then contain the following series of data records:

Record	Contents	Example
1	#Table Index	#Table Index 0x0030
2	32 bit hexadecimal word #1 in VM table	0xa0820000
3	32 bit hexadecimal word #2 in VM table	0x0020c800
4	32 bit hexadecimal word #3 in VM table	0xa0820001
.	.	.
.	.	.
.	.	.
N+1	32 bit hexadecimal word #N in VM table	0x00000002

- The Table Index record will begin with a '#' symbol.
- The Table Index will denote the index value of the 32 bit hexadecimal word in the line following the Table Index record.
- The Table Index will normally be written in hexadecimal notation, preceded by a '0x'. If it is not preceded by a '0x' then it should be treated as a decimal index.
- There should be no leading or trailing blanks or other characters (e.g. commas, periods, etc) in these records.

6. VM TABLE FILE DATA RECORD BLOCKS

The block defined in section 5 may be repeated a number of times in the VM table file.

7. VM TABLE FILE TRAILER RECORDS

The end of a VM Table file will be denoted by an End Of File marker. No other trailer records are defined following the blocks of standard data records defined in sections 5 and 6.



Appendix – Example VM Table File

```
#CLname: CHOP
#VMVersion: 1.2
#CLVersion: 2.1
#CLCVSId: 24678
#TableId: 0x50
#TableLength: 0x28
#Date: 031201123456
#Table Index 0x0000
0x1200000a
0x00000022
0x120000ff
0x00000000
0x1200000f
0x00000000
0x20100000
0x080186a0
0x02000000
0x02000000
0x12000011
0x00000004
0x4c11000a
0x1000000a
0x12000011
0x00001234
0x4c11000a
0x1000000a
0x4c0f000a
0x1000000f
0x1000000a
0x4c10000a
0x21101001
0x1000000a
0x12000011
0x00005678
0x4c11000a
0x1000000a
#Table Index 0x001c
0x1200000a
0x00000022
0x61660a05
0x54000066
0x02000000
0x30ffffe9
0x00000000
0x00000000
0x00000000
0x00000000
0x00000000
0x00000000
0x00000000
0x00000000
0x7f000000
```