Vanilla Examples with the CIRS database

Conor A. Nixon
University of Maryland/ NASA GSFC

1 Introduction

This document gives examples of examples of UNIX command-line statements for extracting CIRS data using the Vanilla program. Vanilla is a simple database query and extraction tool, originally designed for use with the Mars Global Surveyor (MGS) Thermal Emission Spectrometer (TES) instrument data set, but now coverted for Cassini CIRS use. Note that the CIRS implementation of Vanilla is no longer fully compatible with the ASU/MGS-TES implementation.

1.1 Scope

This document is not intended to be an implementation or data-creator's guide to Vanilla: see the *ASU Vanilla User's Guide*. It is also not a complete reference guide to the CIRS database, especially fields. See the `CIRS Data Product Software Interface Specification` CIRS Data SIS.

1.2 Vanilla usage

The general form of the Vanilla command is:

```
vanilla <dir> <output_fields_string> <selection_criteria_string>
```

The three arguments may occur in any order. The `<output_fields_string>` takes the following form:

```
-fields "fieldname1 fieldname2 fieldname3 ...  "
```

and the `<selection_criteria_string>` takes the following form:

```
-select "fieldname1 minrange1 maxrange1 fieldname2 minrange2 maxrange2 ...  "
```

Note that the trailing '.' in the examples *is* important: it indicates the directory to be searched, and stands for the current directory. This may be replaced by any other directory path name which contains a DATASET.TXT file. The example output was produced by running the commands in the `/DATA/APODSPEC` directory.

1.3 SCET times

In nearly all cases of querying the database, the user will wish to restrict the range of times in the sample. The CIRS Vanilla database is ordered first and foremost in SCET time, as explained in the Data SIS document. To convert a simple text time into integer SCET (seconds from 1970 00:00:00), there exists a tool called `cirs_time` on every archive volume. Example:

```
cirs_time 12/31/00 00:00:00
```

prints '978220800', and the reverse:

```
cirs_time 978220800
```

prints:

(DOY 366) Sun Dec 31 00:00:00 2000.

2 Examples

2.1 Example 1

To request SCET times, detector numbers, RTI lengths, and number of data points for all spectra in the directory `/DATA/APODSPEC`:

```
vanilla -fields "scet det rti ispts" /DATA/APODSPEC
```

Output:

```
scet det rti ispts
970538722 21 40 112
970538722 22 40 112
970538722 23 40 112
970538722 24 40 112
970538722 25 40 112
970538727 3 40 112
970538727 4 40 112
970538727 5 40 112
970538727 6 40 112
...
(output truncated)
```

2.2 Example 2

To request spectra for detector 0 only (FP1) between midnight January 1st 2001 and midnight 23 hours later, querying the current directory:

```
vanilla -fields "scet ispts ispm[1:2]" -select "scet 978307200 978393600 det 0 0" .
```

Output:

```
scet    ispts   ispm[1] ispm[2]
978317375       691     2.4352957428E-09        1.7582391099E-09
978317387       691     1.1043535864E-08        -5.2629736125E-09
978317399       691     2.2205675521E-08        1.1947789424E-08
978317411       691     8.0703310701E-09        1.4675430648E-08
978317423       691     -4.5585211161E-09       -6.2420228986E-09
978317435       691     2.9919753342E-09        6.2782912202E-09
978317447       691     2.1354420454E-09        -2.1846500164E-09
978317459       691     -2.743928175E-11        3.5971765566E-09
978317471       691     3.0526297046E-09        2.7907505196E-09
...
(output truncated)
```

Note that the contents of the square brackets '1:2' specify in this case only to extract the first two points in each spectrum. This range may be modified, or left blank, in which case the entire spectrum is returned. If the square brackets are absent, the byte offset of the spectrum in the variable length datafile is returned.

2.3 Example 3

To request Jupiter spectra for FP3 detector 5 (including limbs), reading from the current directory.

```
vanilla -fields "scet ispts ispm[1:2]" -select "fov_targets 2 3 det 5 5" .
```

Output:

```
scet ispts ispm[1] ispm[2]
970538727 112 1.5121288754E-07 1.7017845266E-07
970538737 112 5.7385352648E-07 4.0110046484E-07
970538747 112 1.5460256009E-07 1.7092061455E-07
970538757 112 5.7628477634E-07 3.942838589E-07
970538767 112 1.3852478276E-07 1.52706221E-07
970538777 112 6.4783472453E-07 4.6467275183E-07
970538787 112 1.7820106279E-07 1.729755752E-07
970538797 112 5.5494604112E-07 3.8624628473E-07
970538807 112 1.3961407319E-07 1.547202686E-07
...
(output truncated)
```

2.4 Example 4

To request FP1 polarizer temperatures and SCET times, re-directing the output to a file 'vanilla.txt'.

```
vanilla -fields "scet firpolriztem" .  > vanilla.txt
```

The file vanilla.txt contains:

```
scet firpolriztem
970538722 170.30636337
970538722 170.30636337
970538722 170.30636337
970538722 170.30636337
970538722 170.30636337
970538727 170.30657246
970538727 170.30657246
970538727 170.30657246
970538727 170.30657246
...
(output truncated)
```

2.5 Example 5

To request all observations where the center of the FOV is on Jupiter, printing observation times, detector numbers, and center latitudes of each FOV:

```
vanilla -fields "scet det latitude_zpd[5]" -select
"altitude_zpd[5] 0.0 0.0 fov_targets 2 3" .
```

Output:

```
scet det latitude_zpd[5]
970538722 22 55.700315033
970538722 23 54.351969487
970538727 3 55.082600148
970538727 4 55.839988687
970538727 5 55.168586613
970538727 6 53.012304072
970538732 4 55.839844933
970538732 6 53.013129464
970538737 3 55.081237136
...
(output truncated)
```

2.6 Example 6

As in example 5, but also requiring only FP3 detectors, that each detector is entirely on the disk of the planet, and emission angles less than 60 degrees (average for FOV).

```
vanilla -fields "scet det latitude_zpd[5] emission_angle[5]" -select "fov_targets 2 3
all_q_on 1 1 det 1 10 emission_angle_fov_average 0.0 60.0" .
```

Output:

```
scet    det     latitude_zpd[5] emission_angle[5]
978317771       6       58.976467728    59.739055311
978317783       3       57.282600517    58.75512698
```

```
978317783       5       57.268189011    58.071412216
978317783       7       57.27026991     58.163721553
978317783       9       57.288978225    59.035901106
978317795       2       55.77162325     57.798991895
978317795       4       55.75096519     56.764020657
978317795       6       55.745752985    56.496227335
978317795       8       55.755646898    56.986041772
...
(output truncated)
```

2.7 Example 6

A tricky situation may arise when extracting information from the GEO tables, which contain spacecraft position relative to major bodies in the system. For the Jovian system, this means Jupiter and the Galilean satellites.

Consider the request:

```
vanilla -fields "scet det latitude_zpd[5] body_spacecraft_range" -select "fov_targets
2 3 all_q_on 1 1 det 5 5" .
```

where `body_spacecraft_range` is the field from the GEO tables. The output begins:

```
scet    det     body_spacecraft_range
978317399       5       9730206.0155
978317399       5       10307398.3
978317399       5       9472648.4923
978317399       5       10022884.398
978317399       5       9926768.1554
978317663       5       9726715.9272
978317663       5       10305092.328
978317663       5       9470760.952
978317663       5       10024941.492
978317663       5       9927243.111
...
(output truncated)
```

Notice how there are apparently five different ranges at each SCET time, even though we asked for just spectra when Jupiter/and rings were in the FOV, not satellites. This is because the ranges to the major satellites are always produced in the tables, whether they are visible or not. I.e., there is a five-fold degeneracy specified by an additional field: `body_id`. We can add this field explicitly to the output fields to show the degeneracy:

```
vanilla -fields "scet det latitude_zpd[5] body_spacecraft_range body_id" -select "fov_targets
2 3 all_q_on 1 1 det 5 5" .
```

produces:

```
scet    det     body_id body_spacecraft_range
978317399       5       501     9730206.0155
```

```
978317399       5       502     10307398.3
978317399       5       503     9472648.4923
978317399       5       504     10022884.398
978317399       5       599     9926768.1554
978317663       5       501     9726715.9272
978317663       5       502     10305092.328
978317663       5       503     9470760.952
978317663       5       504     10024941.492
978317663       5       599     9927243.111
...
(output truncated)
```

To remove the degeneracy, the value of `body_id` can be restricted explicitly in the 'select' part of the query string.

Note that there can be a similar degeneracy in the POI tables, when multiple targets appear in the FOV. Then the degeneracy is in the `target_id` parameter.