Note 212: Basic Flat Table to MeasurementSet Conversion

J.P. McMullin
jmcmulli@nrao.edu

Sept 22, 1997

1 Motivation

In anticipation of switching from flat tables to MeasurementSets, this note is intended to describe the alterations needed in existing scripts to accomodate this change.

2 Introduction

In construction, a measurement set is fashioned almost identically to a flat table (see the gbt-module, gbtfiller and gbtmsfiller). All of the information contained within the flat tables is also within a MeasurementSet, however, the information is reorganized which necessitates changes in existing scripts to access the data, columns and cells. (See Note 191 for complete information on MeasurementSet definitions).

3 Basic Mapping

This section presents a simple comparison key to accessing information in flat tables versus a MeasurementSet. The following table summarizes the conversion; the Flat Table Column is the table keyword used in flat tables and the MeasurementSet Column is the corresponding keyword while Table specifies whether the information exists in the main table or in a subtable. Data information in the "main" table is invoked normally:

```
- a := table('example.MS');
- scans := a.getcol('SCAN_NUMBER');
```

Data information in subtables is accessed by adding a subdirectory to the MeasurementSet descriptor:

```
- a := table('example.MS/SOURCE');
- object := a.getcol('NAME');
```

Flat Table Column	MeasurementSet Column	Table
Time	TIME[a]	main
DATA	FLOAT_DATA[b]	main
SCAN	SCAN_NUMBER	main
SUBSCAN	NS_GBT_SUBSCAN	main
RECEIVER_ID	RCVRID	NS_GBT_BACKEND
OBJECT	NAME	SOURCE
PSRDM	PSRDM	NS_GBT_BACKEND
PHASE_ID	NS_GBT_PHASE_ID	main
UTCSTART	TIME[a]	main
UTDATE	TIME[a]	main
PHASETIM	PHASETIM	NS_GBT_BACKEND
FREQRES	RESOLUTION	SPECTRAL_WINDOW
OBSFREQ	REF_FREQUENCY	SPECTRAL_WINDOW
RFSIDE	RFSIDE	NS_GBT_BACKEND
IFSIDE	IFSIDE	NS_GBT_BACKEND
IFF	IFF	NS_GBT_BACKEND
CAL	CAL	NS_GBT_BACKEND
SIGREF	SIGREF	NS_GBT_BACKEND
PHASETIM	PHASETIM	NS_GBT_BACKEND
BLANKTIM	BLANKTIM	NS_GBT_BACKEND

[a] In MeasurementSets, the TIME is in UTC seconds. To convert to flat table Time:

```
- a := table('example.MS'); # Get the table into Glish.
- b := a.getcol('TIME'); # Get the TIME column.
- c := dm.unit(b[1],'s'); # Add the units explicity.
- d := dm.epoch('utc',c) # Define the epoch.
- e := d.m0.value # e will be the 'Time' from the flat table.
```

To get UTDATE, use the 'r_utc' reference code in the dm.epoch command.

[b] DATA is a 2-dimensional array while FLOAT_DATA is a 3 dimensional array. (e.g. if data::shape is [256,1357], then float_data::shape will be [1,256,1357].

4 Indexing Issues

The main table and subtables within a MeasurementSet may not have the same number of rows. As a result, accessing pieces of information in different subtables will sometimes require more complicated indexing than in a standard flat table.

In general, this can be treated using the table 'query' method to create subtables based on indexing expressions.

For example,

-

It is useful to note that two of the major tables for data analysis, the main table and NS_GBT_BACKEND do have the same number of rows so the indexing is trivial in this case.