ISDAT 2.0 Definition of Filters

Editors:
Anders Lundgren and Gunnar Holmgren
Swedish Institute of Space Physics
Uppsala Division
S-755 91 Uppsala
Sweden

November 24, 1999

Document Status Sheet			
1. Document Title: **ISDAT 2.0 Filters**			
2. Document Reference Number: **CWD-IRF-TN-001**			
3. Issue	4. Revision	5. Date	6. Reason for Change
Draft	0	94 Nov 14	New document

Contents

1 Introduction

1.1 Purpose of the document

This document describes the design of ISDAT filters. It is intended for use within ISDAT software development teams.

1.2 Definition and scope of ISDAT filters

The scope of the *ISDAT filters* is to provide a mechanism for data screening or manipulation to be used by ISDAT clients. By *filters*[1] we mean general purpose software that can be called from ISDAT clients in order to manipulate data in a standard manner. Examples of such manipulation might be digital frequency filters, FFT, merging of data sets, simple arithmetic operations, feeding data to a file or printer, reading of data from a file etc.

1.3 Acronyms, and abbreviations

The used acronyms and abbreviations are explained in Table 1.

Acronym	Meaning
ISDAT	Interactive Science Data Analysis Tool
TBD	To be defined
TBW	To be written
WEC	Wave experiment Consortium

Table 1: Acronyms and abbrieviations

2 Reference Documents

[1] C. Morris (Editor). *Academic Press Dictionary of Science and Technology.* Academic Press, 1992.

[2] G. Holmgren and A. Lundgren. WEC detailed data analysis software. architectural design. Technical Report CWD-ADD-001, IRF-U, February 1994.

2.1 Overview of the document

Section 3 of this document briefly identifies the requirements for filters, and section 4 defines the interfaces. In Appendix A an example of a filter source code is given.

[1] According to [Ref. 1] "filter is any device or process that serves to screen out something".

3 Requirements on Filters

For WEC applications, the following filters are planned, [Ref. 2]:

fileFilter For export or import data to/from files.

CoordFilter for coordinate transformations.

averFilter for averaging of data and removing of dc-levels.

analFilter for simple arithmetics between parameters.

unitsFilter for units conversions.

modulusFilter for computation of modulus.

traceFilter

joiningFilter for joining of data sets.

This may be translated into the following software requirements:

- One or several data sets must be available to the filter.
- The number of returned data sets may differ from the number of used data sets.
- Much, if not all, meta data must be available to the filter.
- The filter must be allowed to change the meta data.
- Additional "steering" parameters must be communicated to the filter.
- The filter must be able to return a success/error flag.
- The filter must be able to return an error/varning code.
- The filter must be able to return an error/warning string.
- The filter must be allowed to have its own graphical user interface (window).

4 Interface descriptions

The interface between the *filters* and the ISDAT is defined by the calls described in the following sections.

4.1 IsFilter

4.2 IsCallFilter

4.3 IsFilterRead

4.4 IsFilterWrite

4.5 IsRegisterFilter

4.6 IsUpdateFilterStruct

4.7 Filter data structure

```
typedef struct _IsFilterDesc {
    DbDataObject objectList;    /* A list of pointers to objects */
                                /* Filters are allowed to modify */
    int numberOfObjectsToFilter;
    int numberOfObjectsFromFilter;
    int intParmeter[5];  /* User defined parameters, in and out */
    float floatParameter[5]; /*User defined parameters, in and out*/
    int errorCode;
    char *string;     /* error/warning string */
} IsFilterDesc;
```

A Example of filter source code

[This section will give an example of a filter source code, TBW]