Open Data Interface - SAAPS/SEDAT/SPENVIS
(ODI-SSS)
Database Technical Note

Peter Wintoft
Swedish Institute of Space Physics
and
Daniel Heynderickx
DH Consultancy

September 25, 2009

Swedish Institute of Space Physics

DH Consultancy
Document status sheet

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>September 25, 2009</td>
<td>First version.</td>
</tr>
</tbody>
</table>
Contents

1 Summary 4
2 Abbreviations 4
3 Introduction 4
4 SAAPS 5
   4.1 SAAPS 1.2 ..................................................... 5
   4.2 SAAPS 2.0 ..................................................... 5
      4.2.1 The saaps.dbt.ODIDatabase class ....................... 5
      4.2.2 Implementation ......................................... 7
5 SEDAT 7
6 SPENVIS 7
1 Summary

This document describes the adaption of SAAPS, SEDAT, and SPENVIS to operate against the ODI database.

2 Abbreviations

ACE  Advanded Composition Explorer
ASCII  American Standard Code for Information Interchange
CDF  Common Data Format
GOES  Geostationary Operational Environment Satellite
HTTP  Hypertext Transfer Protocol
ITT/IDL  ITT Interactive Data Language
ODI  Open Data interface
SAAPS  Satellite Anomaly Analysis and Prediction System
SEDAT  Space Environment Data Analysis Tool
SOW  Statement of Work
SPENVIS  Space Environment Information System
SQL  Structured Query Language
TSO  Time Series Object

3 Introduction

The Open Data Interface (ODI) is a database system that will be accessible and store data from SAAPS, SEDAT, and SPENVIS systems. As ODI is based on MySQL the system is accessible from other systems too. Interfaces to languages like C, Java, PHP, IDL, Matlab exist.

One of the requirements on ODI is that it must be able to store data compliant with the CDF/ISTP/PRBEM guidelines. ISTP defines a set of metadata that must be present to describe solar-terrestrial physics data. ODI is constructed so that it can hold all the ISTP metadata and therefore becomes a general system for storing solar-terrestrial physics data. ODI makes use of the CDF/ISTP metadata definition, seen e.g. in the CDF skeleton files, to populate the ODI data tables holding the metadata. The same approach is also used for non-CDF files, like ASCII files. The PRBEM guidelines
extend the ISTP guidelines by defining a set of variables relevant for radiation belt data. The PRBEM extension does not affect the ODI database model.

In the following sections we describe the underlying ISTP metadata, how it is stored in ODI, how tables to hold metadata and data are created, how data are ingested and exported.

4 SAAPS

The latest version of SAAPS that works against the SAAPS database is 1.2. In this work SAAPS 1.2 is developed to work against the ODI database. This version is called SAAPS 2.0.

SAAPS 1.2 was a complete system consisting of both tools and database. The database was updated with real time data. In SAAPS 2.0 only the tools are needed as the ODI system keeps the database up to date.

4.1 SAAPS 1.2

SAAPS is implemented in Java and is organised into three basic packages according to Table 1, and a few additional support packages.

<table>
<thead>
<tr>
<th>Package name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>saaps.dbt</td>
<td>The database tools to access data from the database.</td>
</tr>
<tr>
<td>saaps.saam</td>
<td>The satellite anomaly analysis module.</td>
</tr>
<tr>
<td>saaps.sapm</td>
<td>The satellite anomaly prediction module.</td>
</tr>
</tbody>
</table>

The entry point to access the database in SAAPS 1.2 is the class saaps.dbt.Database. This class provides methods to connect to the database, read and write data, and read and write metadata.

4.2 SAAPS 2.0

In SAAPS 2.0 the saaps.dbt.Database class is replaced by saaps.dbt.ODIDatabase that keeps the same functionality for reading but instead connects to the ODI database. The writing methods are not implemented as they are not used. All classes that previously used saaps.dbt.Database are updated to use saaps.dbt.ODIDatabase and they are listed in Table 2.

4.2.1 The saaps.dbt.ODIDatabase class

The saaps.dbt.ODIDatabase class provides methods for connecting and reading data from the ODI database. A connection is made according to
Table 2: All classes that are updated to use `saaps.dbt.ODIDatabase`.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>saaps.dbt.DatabaseItems</code></td>
<td>A class to hold the names of the datasets.</td>
</tr>
<tr>
<td><code>saaps.dbt.ShowMetaData</code></td>
<td>A class to show the metadata relevant for SAAPS.</td>
</tr>
<tr>
<td><code>saaps.saam.DataPlotterApp</code></td>
<td>Tool to plot data from the ODI datasets.</td>
</tr>
<tr>
<td><code>saaps.saam.ElFluence</code></td>
<td>A class used to extract daily electron fluences from GOES. Used by the Electron Fluence Levels tool in the Analysis Module.</td>
</tr>
<tr>
<td><code>saaps.saam.SEAApp</code></td>
<td>A tool to perform Superposed Epoch Analysis using data from the ODI database.</td>
</tr>
<tr>
<td><code>saaps.saam.SumKp</code></td>
<td>A class to extract the daily sums of Kp over a period. Used by the Sum Kp Level tool of the Analysis module.</td>
</tr>
<tr>
<td><code>saaps.sapm.Anom001Model</code></td>
<td>A model for predicting anomalies for satellite type 001.</td>
</tr>
<tr>
<td><code>saaps.sapm.Anom002Model</code></td>
<td>A model for predicting anomalies for satellite type 002.</td>
</tr>
<tr>
<td><code>saaps.sapm.Anom003Model</code></td>
<td>A model for predicting anomalies for satellite type 003.</td>
</tr>
<tr>
<td><code>saaps.sapm.Anom004Model</code></td>
<td>A model for predicting anomalies for satellite type 004.</td>
</tr>
<tr>
<td><code>saaps.sapm.ElFluence</code></td>
<td>A class to compute the daily electron fluence from GOES.</td>
</tr>
</tbody>
</table>
ODIDatabase db = new ODIDatabase()

which will connect to the database at user level 3 based on the ODI environment variables. See TN ODI.

The basic method to read data is

TimeSeries ts = db.retrieveTimeSeries(String dataset, Date start, Date end)

where TimeSeries is a SAAPS class to handle time series data.

4.2.2 Implementation

For SAAPS to function the following environment variables needs to be set:

$SAAPS_DIR : The directory where SAAPS is installed.
$SAAPS_HOME : The directory where the SAAPS executables are installed.
$SAAPS_LIB : The directory of external packages.

On spitfire.estec.esa.int these have been set to

SAAPS_DIR=$ODI_HOME/SAAPS
SAAPS_HOME=$SAAPS_DIR
SAAPS_LIB=$SAAPS_DIR/lib

To run SAAPS the following is executed on the command line

$SAAPS_HOME/saaps.sh

which will start the SAAPS user interface.

5 SEDAT

[ To be written. ]

6 SPENVIS

[ To be written. ]