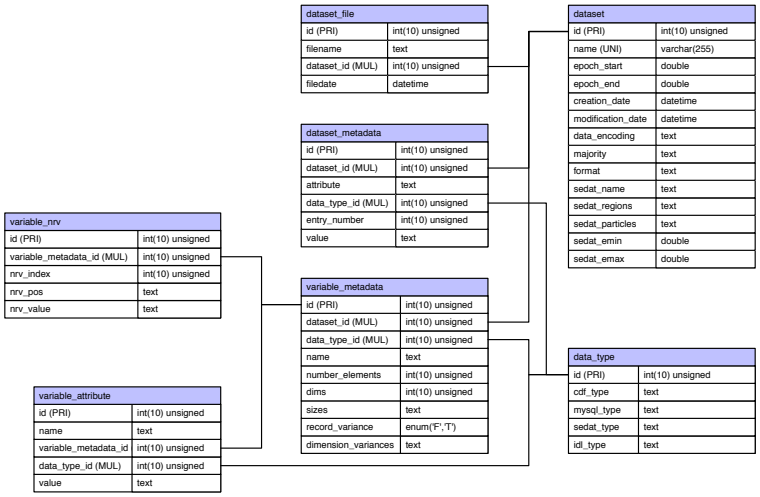
ODI Administrator Guide
ESTEC/Contract No. 21964/08/NL/AT

September 25, 2009



Peter Wintoft, Swedish Institute of Space Physics

Daniel Heynderickx, DH Consultancy

Document status sheet

Version	Date	Comment
1.0	2009-06-01	First version.
1.1	2009-09-25	Updated.

Contents

1 Summary

The Open Data Interface (ODI) database builds on the MySQL database server to store space physics data compliant with the CDF/ISTP/PRBEM guide lines. In this manual we describe how to set up the database and how to ingest data.

2 Prerequisites

2.1 Software

To run ODI the following must be installed on the server:

- MySQL 5.0 or higher.
- PHP 5.2 or higher.
- Access to the MySQL client library.
- wget

2.2 Database and users

A MySQL database must also be created with three levels of users with different privileges. In the following we assume the database is named `odi`. The three users levels are

User Level 1 (UL1) which has all privileges on the database,

User Level 2 (UL2) which has select, update, and insert privileges,

User Level 3 (UL3) which has only select privileges.

The UL1 user is typically an administrator user of the database. The UL2 user is used by the automatic routines that updates the database. Finally, the UL3 user is the normal end user or application who only needs to pull data out of the system. The names and passwords of each UL are specified in the environment variables below.

2.3 Enironment variables

A number of environment variables must also be set up:

`$ODI_HOME`	The base directory where ODI is installed.
`$ODI_DATA`	The base directory where the imported files are stored.
`$ODI_HOST`	The host name of the MySQL server.
`$ODI_PORT`	The port number through which to communicate with the MySQL server.
`$ODI_SOCKET`	The MySQL socket.
`$ODI_USER_1`	The name of the User Level 1 user. This user has all privileges on the tables in the ODI database.
`$ODI_PW_1`	The password of the User Level 1 user.
`$ODI_USER_2`	The name of the User Level 2 user. This user has the privileges to read data from and write data to the ODI database.
`$ODI_PW_2`	The password of the User Level 2 user.
`$ODI_USER_3`	The name of the User Level 3 user. This user has only privileges read data from the ODI database.
`$ODI_PW_3`	The password of the User Level 3 user.

3 Installing the ODI software

A compressed tar archive exists at `http://www.lund.irf.se/odi/internal` which can be downloaded and extracted, which will create files in the following directories:

`doc`	Various documentation, e.g. this page.
`lib`	ODI library code.
`parsers`	Code to parse the raw datasets and ingest them into ODI.
`readme.txt`	Various information.
`SAAPS`	The SAAPS system.
`SEDAT`	The SEDAT system.
`setup_database`	Code to create the core ODI database tables.
`SPENVIS`	The SPENVIS system.
`test`	Various test code.
`tools`	Various tools to use on the ODI database.

4 Creating the ODI database

Before ODI is installed the database must contain no data. If `odi` was just created this will be the case, but if not any possible tables may be removed using `drop_all_tables.php`. To create the core ODI tables the following is run

```
>> cd $ODI_HOME/setup_database
>> drop_all_tables.php
>> create_odi_database.php
```

where >> means the Unix shell and assuming that the current directory (".") is in the search path. This will create 7 tables which are shortly described below and illustrated in Figure 1.

`dataset`

The table dataset contains global information of all datasets that exist in ODI and is the top level description of a dataset in ODI. This includes the name of the table holding the data for a specific dataset. The names of the data tables always start with `dataset_` followed by a unique name, e.g. `dataset_rosetta_srem_pacc`. The information in the CDF header block also goes into dataset, namely: `DATA ENCODING`, `MAJORITY`, and `FORMAT`. The `CDF NAME` is not stored as it changes when new files are ingested into the dataset. However, `CDF NAME` can be recreated from the information stored in the database.

`dataset_metadata`

The table `dataset_meta` contains the next level of metadata for each dataset. This metadata comes from the global attributes block in the CDF skeleton: `Attribute Name`, `Entry Number`, `Data Type`, and `Value`.

`variable_metadata`

The table `variable_metadata` contains the metadata for each variable. In the CDF skeleton each z-Variable is defined by `Variable Name`, `Data Type`, `Number Elements`, `Dims`, `Sizes`, `Record Variance`, and `Dimension Variances`. All this information goes into `variable_metadata`.

`variable_attribute`

To each variable in the CDF dataset there are a varying number of associated attributes. The `Attribute Name`, `Data Type`, and `Value` for each attribute are stored in the table `variable_attribute`.

`variable_nrv`

Any data that does note vary from record to record has the CDF "Record Variance"=F(alse) and is known as non-record-variant (nrv) data. The nrv values appears in the skeleton file and are stored in the table `variable_nrv`.

`data_type`

The table `data_type` contains all CDF data types and associated MySQL and SEDAT data types.

`dataset_file`

The table `dataset_file` stores the names and dates of all files that have been ingested into ODI.

`variable_depend`

The table `variable_depend` stores the dependencies between variables as identified by the CDF variable attribute `DEPEND_i`.

At this stage, the only table that contains data is the `data_type` table. The other tables will later on be populated with various metadata.

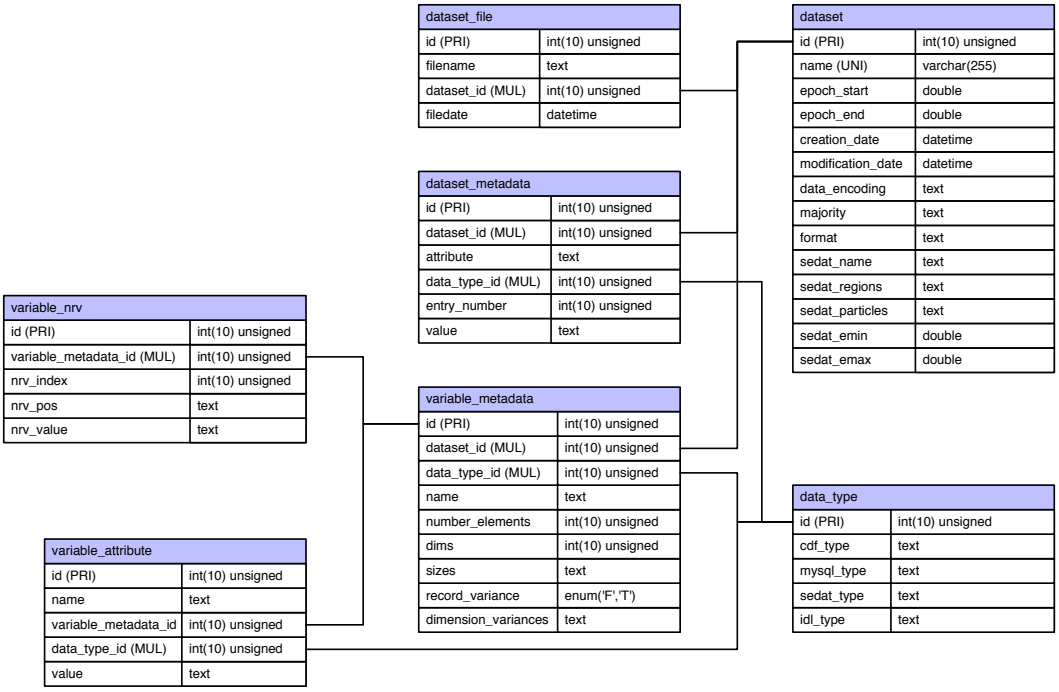


Figure 1: The ODI metadata tables.

5 Ingesting data

The procedure to ingest data is to execute the `populate.php` program, in `$ODI_HOME/parsers/`, which will read the file `datasets.txt`.

```
>> cd $ODI_HOME/parsers
>> populate.php [<dataset 1> [<dataset 2> ... ] ]
```

In order for this to work a number of files are needed. Firstly, the *dataset definition file* and the *skeleton file* must exist. Then the *raw data files* must exist, either locally or remotely. If the raw data files are CDF files then there must also exist a CDF *settings file*.

The dataset definition file, named `datasets.txt`, contains information about the dataset not contained in the skeleton file. The structure of the `datasets.txt` file is

```
<ODI data table name>;;; <Data directory>;;; <File name pattern>;;;\
 <Platform>;;; <Platform type>;;; <Instrument>;;; <Skeleton file name>;;;\
 <SEDAT dataset name>;;; <SEDAT region code>;;; <SEDAT particle code>;;;\
 <SEDAT min. energy (MeV)>;;; <SEDAT max. energy (MeV)>
```

where the row has been split over several lines to fit into this document. Each field is separated with a triple-colon (`;;;`) and the text within angle brackets (`<>`) should be replaced with actual values. This file is parsed and the value of the field `<ODI data table name>` is stored in `dataset.name`.

The `<Data directory>` gives the location of the data files relative to `$ODI_RAWDATA`. The actual data files must not necessarily be placed directly under `<Data directory>` but can be placed in subdirectories.

The data files that are to be ingested must match the `<File name pattern>` field. The `<File name pattern>` may contain the %-sign which is treated as a wild card and will match any string.

The `<Platform>`, `<Platform type>`, and `<Instrument>` are stored in the `dataset_metadata` table together with the other global attributes for the dataset.

The `<Skeleton file name>` field gives the name of the skeleton file that shall be used.

The SEDAT fields are stored in the table `dataset` in the columns `sedat_name`, `sedat_regions`, `sedat_particles`, `sedat_emin`, and `sedat_emax`.

When all files are present the `populate.sh` script is executed, which will create dataset tables when necessary and ingest the data.

When `populate.php` is executed without any arguments all lines in the `datasets.txt` file that do not start with a `#`-character are parsed. If arguments like

```
populate.php <dataset 1> <dataset 2> ...
```

are passed only the lines with those datasets will be parsed, irrespective of whether there is a leading `#`-character.

When the raw data comes from CDF files they are automatically converted to text files with comma-separated fields using the CDFexport program. The text files are ingested into the `dataset_` table. It is **very important to order the data fields**

correctly in the text files, which is controlled by the settings file (`.set`), to match the order of the columns in the `dataset_*` table. The columns in the `dataset_*` table have the same order as the variables are defined in the skeleton file.

When the raw data is a text file there are so many different possibilities how it is formatted. Therefore, to ingest text data the `$ODI_HOME/lib/odi.parsers.php` must be edited. To give an example, the following code section in `odi.parsers.php` ingests real time Dst data from Kyoto and Lund.

```
// Add the Kyoto HTML pages and Lund forecast page to the file list for Dst
    if ($filepattern == "%dst%")
    {
      $dbasedir = "http://swdc234.kugi.kyoto-u.ac.jp/dst_realtime";
      $findex = file($dbasedir . "/index.html");
      foreach ($findex as $dumstr)
      {
        if (($di = strpos($dumstr, "/index.html")) &&
            (strpos($dumstr, "month") === FALSE) &&
            (strpos($dumstr, "dst") === FALSE))
        {
          $file = $dbasedir . substr($dumstr, $di-7, 18);
          $dum = file($file);
          foreach ($dum as $line)
          {
            $offset = strpos($line, "[Updated");
            if ($offset !== FALSE)
            {
              $date = explode("UT", substr($line, $offset+11));
              $date = date("Y/m/d H:i:s", strtotime($date[0] . ":00"));
              break;
            }
          }
          $dfiles[] = array($file, $date, $file);
        }
      }
      $dfiles[] = array("http://rwc.lund.irf.se/rwc/dst/dst1.txt",
                        date("Y/m/d H:i:s"),
                        "http://rwc.lund.irf.se/rwc/dst/dst1.txt");
    }
```

If the dataset does not already exist in the ODI database the `datasets.txt` and skeleton files will be parsed and the data will be stored in the ODI metadata tables. Then the corresponding `dataset_*` table is created. Next, the actual data files, in `$ODI_DATA`, are parsed and ingested into the `dataset_*` table. However, a check is first made on the data file name to see whether it has already been stored in ODI and if it has no parsing or ingesting will be performed. The procedure is illustrated in Figure 2.

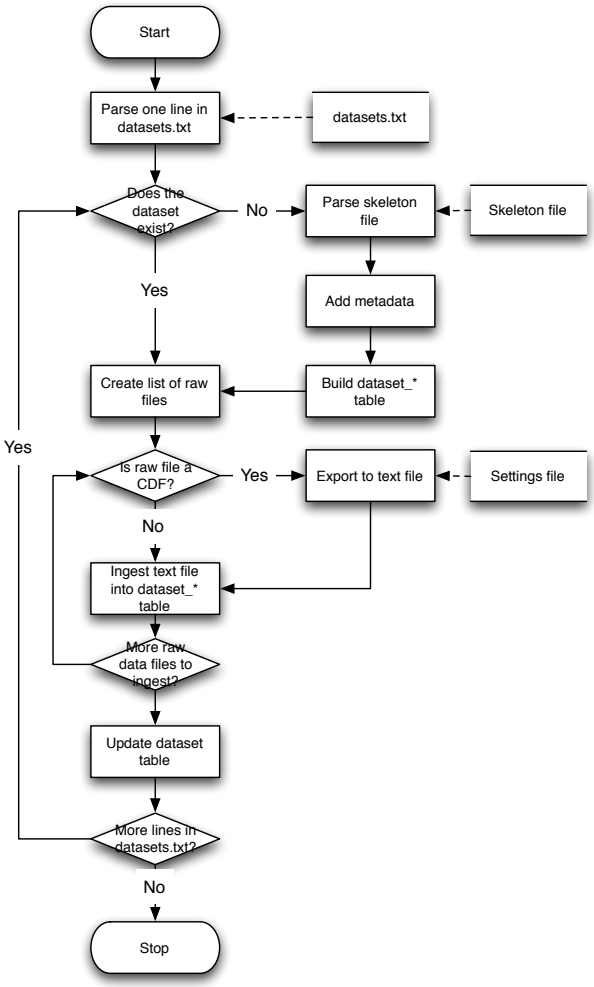


Figure 2: The diagram illustrates the procedure to ingest data into ODI.

To summarise, the following steps must be performed to add a new dataset to the ODI database:

1. Add a row to the `datasets.txt` file;
2. Add a directory under `$ODI_DATA` with subdirectories according to `<Data directory>` in `datasets.txt`.
3. In `$ODI_HOME/parsers` create a directory name `<platform>`.
4. Add the file `<instrument>.skt` that contain the dataset metadata according to CDF skeleton syntax.
5. Then depending on whether it is a text file or CDF file to be ingested do:

Text Edit the `$ODI_HOME/lib/odi.parsers.php` for the parsing of the text file.

CDF Add the file `<instrument>.set` that contain the CDF export configuration.

6. Run `popoulate.php [<dataset>]`.

6 Updating the database with live data

It is assumed that the ODI metadata tables have been populated with metadata and that the associated `dataset_*` table has been created as described in the previous section. Then an entry is added to the cron-table using the Unix `crontab` program. In `$ODI_HOME/lib/` the file `cron_jobs.txt` contains a set of datasets that are updated with live data. This file may be changed to add more entries. To start the jobs execute

```
>> cd $ODI_HOME/lib
>> crontab cron_jobs.txt
```

7 Command line tools

Under the directory `$ODI_HOME/tools/` there are a set of programs that can be used on the ODI database.

7.1 `check_skeleton.php <skeleton file>`

For ODI to operate correctly there are certain set of metadata that need to be defined in the skeleton file. The program `check_skeleton.php` parses the skeleton file and reports if there are any missing metadata. The user may then need to edit the skeleton file until it passes the check.

7.2 `get_epoch_range.php <dataset>`

Get the epoch range for the given dataset.

7.3 `show_datasets.php`

Get a list of existing datasets.

7.4 `show_metadata.php <dataset>`

Show the metdata for a selected dataset.

A Listing of `cron_jobs.txt`

```
0,10,20,30,40,50 * * * * /home/odi/odi/live/ace_swepam_download.php \
 > /dev/null
0,10,20,30,40,50 * * * * /home/odi/odi/live/ace_mag_download.php \
 > /dev/null
```

B Datasets included in ODI

Table 1: The table lists all data sets included in ODI. The `dataset_` prefix in the ODI Name is not shown.

ODI Name	SEDAT Name	Description
ace_sis	ACE_SIS	ACE-SIS data
ampte_uks	AMPTE	AMPTE UKS electron data
azur	AZUR	AZUR Proton/Alpha particle telescope data
crres_mea	CRRES	CRRES/MEA data
equator_s_aux	EQUATOR_S_AUX	Equator-S AUX Dataset
equator_s_epi	EQUATOR_S_EPI	Equator-S EPI Dataset
equator_s_mam	EQUATOR_S_MAM	Equator-S MAM Dataset
gioveb_srem_pacc	GIOVEB_SREM_PACC	GIOVE-B/SREM PACC Data
goes_sem_a05_5	SPIDR_GOES_A05_5	SPIDR GOES-5 A dataset 5 Minute resolution
goes_sem_a06_5	SPIDR_GOES_A06_5	SPIDR GOES-6 A dataset 5 Minute resolution
goes_sem_a07_5	SPIDR_GOES_A07_5	SPIDR GOES-7 A dataset 5 Minute resolution
goes_sem_a08_5	SPIDR_GOES_A08_5	SPIDR GOES-8 A dataset 5 Minute resolution
goes_sem_a09_5	SPIDR_GOES_A09_5	SPIDR GOES-9 A dataset 5 Minute resolution
goes_sem_a10_5	SPIDR_GOES_A10_5	SPIDR GOES-10 A dataset 5 Minute resolution
goes_sem_a11_5	SPIDR_GOES_A11_5	SPIDR GOES-11 A dataset 5 Minute resolution
goes_sem_a12_5	SPIDR_GOES_A12_5	SPIDR GOES-12 A dataset 5 Minute resolution
goes_sem_g05_1	SPIDR_GOES_G05_1	SPIDR GOES-5 G dataset 1 Minute resolution
goes_sem_g06_1	SPIDR_GOES_G06_1	SPIDR GOES-6 G dataset 1 Minute resolution
goes_sem_g07_1	SPIDR_GOES_G07_1	SPIDR GOES-7 G dataset 1 Minute resolution
goes_sem_g08_1	SPIDR_GOES_G08_1	SPIDR GOES-8 G dataset 1 Minute resolution
goes_sem_g09_1	SPIDR_GOES_G09_1	SPIDR GOES-9 G dataset 1 Minute resolution

Table 1: (continued)

ODI Name	SEDAT Name	Description
goes_sem_g10_1	SPIDR_GOES_G10_1	SPIDR GOES-10 G dataset 1 Minute resolution
goes_sem_g11_1	SPIDR_GOES_G11_1	SPIDR GOES-11 G dataset 1 Minute resolution
goes_sem_g12_1	SPIDR_GOES_G12_1	SPIDR GOES-12 G dataset 1 Minute resolution
goes_sem_h06_5	SPIDR_GOES_H06_5	SPIDR GOES-6 H dataset 5 Minute resolution
goes_sem_h07_5	SPIDR_GOES_H07_5	SPIDR GOES-7 H dataset 5 Minute resolution
goes_sem_h08_5	SPIDR_GOES_H08_5	SPIDR GOES-8 H dataset 5 Minute resolution
goes_sem_h09_5	SPIDR_GOES_H09_5	SPIDR GOES-9 H dataset 5 Minute resolution
goes_sem_h10_5	SPIDR_GOES_H10_5	SPIDR GOES-10 H dataset 5 Minute resolution
goes_sem_h11_5	SPIDR_GOES_H11_5	SPIDR GOES-11 H dataset 5 Minute resolution
goes_sem_h12_5	SPIDR_GOES_H12_5	SPIDR GOES-12 H dataset 5 Minute resolution
goes_sem_i05_5	SPIDR_GOES_I05_5	SPIDR GOES-5 I dataset 5 Minute resolution
goes_sem_i06_5	SPIDR_GOES_I06_5	SPIDR GOES-6 I dataset 5 Minute resolution
goes_sem_i07_5	SPIDR_GOES_I07_5	SPIDR GOES-7 I dataset 5 Minute resolution
goes_sem_i08_5	SPIDR_GOES_I08_5	SPIDR GOES-8 I dataset 5 Minute resolution
goes_sem_i09_5	SPIDR_GOES_I09_5	SPIDR GOES-9 I dataset 5 Minute resolution
goes_sem_i10_5	SPIDR_GOES_I10_5	SPIDR GOES-10 I dataset 5 Minute resolution
goes_sem_i11_5	SPIDR_GOES_I11_5	SPIDR GOES-11 I dataset 5 Minute resolution
goes_sem_i12_5	SPIDR_GOES_I12_5	SPIDR GOES-12 I dataset 5 Minute resolution
goes_mag_06	SPIDR_GOES06_MAG	SPIDR GOES-6 MAG dataset 5 Minute resolution
goes_mag_07	SPIDR_GOES07_MAG	SPIDR GOES-7 MAG dataset 5 Minute resolution

Table 1: (continued)

ODI Name	SEDAT Name	Description
goes_mag_08	SPIDR_GOES08_MAG	SPIDR GOES-8 MAG dataset 5 Minute resolution
goes_mag_09	SPIDR_GOES09_MAG	SPIDR GOES-9 MAG dataset 5 Minute resolution
goes_mag_10	SPIDR_GOES10_MAG	SPIDR GOES-10 MAG dataset 5 Minute resolution
goes_mag_11	SPIDR_GOES11_MAG	SPIDR GOES-11 MAG dataset 5 Minute resolution
goes_mag_12	SPIDR_GOES12_MAG	SPIDR GOES-12 MAG dataset 5 Minute resolution
goes_z05_5	SPIDR_GOES_Z05_5	SPIDR GOES-5 Z dataset 5 Minute resolution
goes_z06_5	SPIDR_GOES_Z06_5	SPIDR GOES-6 Z dataset 5 Minute resolution
goes_z07_5	SPIDR_GOES_Z07_5	SPIDR GOES-7 Z dataset 5 Minute resolution
goes_z08_5	SPIDR_GOES_Z08_5	SPIDR GOES-8 Z dataset 5 Minute resolution
goes_z09_5	SPIDR_GOES_Z09_5	SPIDR GOES-9 Z dataset 5 Minute resolution
goes_z10_5	SPIDR_GOES_Z10_5	SPIDR GOES-10 Z dataset 5 Minute resolution
goes_z11_5	SPIDR_GOES_Z11_5	SPIDR GOES-11 Z dataset 5 Minute resolution
goes_z12_5	SPIDR_GOES_Z12_5	SPIDR GOES-12 Z dataset 5 Minute resolution
helios_a_e6	HELIOS_A_E6	HELIOS-A E6 Data
helios_a_e7	HELIOS_A_E7	HELIOS-A E7 Data
helios_b_e6	HELIOS_B_E6	HELIOS-B E6 Data
helios_b_e7	HELIOS_B_E7	HELIOS-B E7 Data
imp8_cpme_e_330s	IMP8_CPME_E_330S	IMP-8 CPME e data
imp8_cpme_h_330s	IMP8_CPME_H_330S	IMP-8 CPME H data
imp8_cpme_he_330s	IMP8_CPME_HE_330S	IMP-8 CPME He data
imp8_cpme_mh_330s	IMP8_CPME_MH_330S	IMP-8 CPME heavy ion data
imp8_crnc_phint	IMP8_CRNC_PHINT	IMP-8 CRNC (U. Chicago) PHINT Data Tape
imp8_gme	IMP8_GME	IMP-8 GME (GSFC Instrument)
index_dst	DST	DST index 1957-1997
index_kpap_1d	AP	Ap global geomagnetic index

Table 1: (continued)

ODI Name	SEDAT Name	Description
index_kpap_3h	KPAP	Kp and Ap global geomagnetic index
index_omni2	NSSDC_OMNI2	NSSDC OMNI-2 Dataset
index_ssn_1m	SSN	Monthly sunspot numbers
integral_irem	INTEGRAL_IREM_PACC	INTEGRAL/IREM PACC Data
isee1_hi	ISEE1_HI	ISEE1 high resolution data
isee1_lo	ISEE1_LO	ISEE1 low resolution data
isee1_mepi	ISEE1_MEPI	ISEE1 MEPI data
isee2	ISEE2	ISEE2 data
meteosat_anomalies	METEOSAT_ANOMALIES	METEOSAT anomalies
meteosat_hr	METEOSAT_HR	METEOSAT high resolution data
meteosat_lr	METEOSAT_LR	METEOSAT low resolution data
metop_02	METOP_02	METOP-02 Space Environment Monitor
mir_a	MIR_A	MIR Raw data
mir_b	MIR_B	MIR reduced and supplementary data
ns41_bdd2r	GPS_NS41	GPS NavStar41 - Burst Dosimeter Detector IIR
poes_n15	POES_N15	NOAA POES N15 Space Environment Monitor
poes_n16	POES_N16	NOAA POES N16 Space Environment Monitor
poes_n17	POES_N17	NOAA POES N17 Space Environment Monitor
poes_n18	POES_N18	NOAA POES N18 Space Environment Monitor
proba1_srem_pacc	PROBA1_SREM_PACC	PROBA-1 SREM PACC Data
rosetta_srem_pacc	ROSETTA_SREM	Rosetta SREM Radiation Monitor
sac_c	SAC_C	SAC-C data
sampex_pet	SAMPEX	SAMPEX PET data
soho_erne_a	SOHO_ERNE_A	SOHO-ERNE Alpha Data
soho_erne_p	SOHO_ERNE_P	SOHO-ERNE Proton Data
strv1b_a	STRV1B_A	STRV1B Raw data
strv1b_b	STRV1B_B	STRV1B reduced and supplementary data

Table 1: (continued)

ODI Name	SEDAT Name	Description
swpc_ace_1m		One minute resolution ACE SWEPAM and MAG data.
uars_pem	UARS	UARS Particle Environment Monitor data
xmm_rm	XMM_RM	XMM Radiation Monitor

C Data tables

dataset	**A table holding the names of each dataset_* table together with some key information.**
id	A unique identifier.
name	The name of the dataset table in the ODI database.
epoch_start	The first epoch in the dataset.
epoch_end	The last epoch in the dataset.
creation_date	The date when the dataset_* table was created.
modification_date	The last date when the dataset was modified.
data_encoding	The CDF header attribute DATA ENCODING.
majority	The CDF header attribute MAJORITY.
format	The CDF header attribute FORMAT.
sedat_name	SEDAT datset name without SYSTEM! prefix.
sedat_regions	SEDAT region code.
sedat_particles	SEDAT particle code.
sedat_emin	SEDAT min. energy (MeV).
sedat_emax	SEDAT max. energy (MeV).

dataset_file	**A table to hold the file names of all ingested data files.**
id	A unique identifier.
filename	The file name of the ingested file.
dataset_id	A key to the associated dataset in table dataset.
filedate	The date when the file was ingested.

dataset_metadata	**Metadata for each dataset.**
id	A unique identifier.
dataset_id	A key to the associated dataset in table dataset.
attribute	The attribute name. This corresponds to the CDF global attribute.
data_type_id	A key to the associated data type in table data_type.
entry_number	The CDF global parameter entry number.
value	The value (or contents) of the dataset attribute.

data_type	**CDF data types together with associated MySQL and SEDAT data types.**
id	A unique identifier.
cdf_type	The CDF data type.
mysql_type	The MySQL data type.
sedat_type	The SEDAT data type.

variable_attribute	**The attributes for each variable.**
id	A unique identifier.
name	The name of the variable attribute.
variable_metadata_id	A key to the associated variable in table variable_metadata.
data_type_id	A key to the associated data type in table data_type.
value	The value (or contents) of the variable attribute.

variable_metadata	**The metadata for each variable.**
id	A unique identifier.
dataset_id	A key to the associated dataset in table dataset.
data_type_id	A key to the associated data type in table data_type.
name	The name of the variable.
number_elements	The CDF variable parameter Number Elements.
dims	The CDF variable parameter Dims.
sizes	The CDF variable parameter Sizes.
record_variance	The CDF variable parameter Record Variance.
dimension_variances	The CDF variable parameter Dimension Variances.

variable_nrv	**The values of the non-record-variant data.**
id	A unique identifier.
variable_metadata_id	A key to the associated variable in table variable_metadata.
nrv_index	An index to the nrv variable. It goes from 1 to dims.
nrv_value	The value of the nrv variable.