ENLIL: Organization and Structure of Data

Dusan Odstrcil

June 11, 2004

Contents

1 Introduction

The purpose of this document is to describe organization and structure of data used and produced by ENLIL code. There are additional documents describing the global context of heliospheric simulations, numerical code, and preparation of input data.

Heliospheric simulation is an activity that requires the user to specify input data, compile the code, run the computation, and process the results. We have adopted various naming conventions to simplify organization and administration of the above work, to facilitate mutual understanding and collaboration, and to enable the development of some auxiliary tools such as data converters and user interfaces.

2 Organization of Data

Basic computational schema is shown in Figure 1. Input data and input parameters are input to the code that produces output data and output log. These files are placed within the directories as shown in Figure 2. System of naming conventions is adopted to identify individual files as shown in Figure 3.

2.1 Project

Computations that are similar are grouped together within a <*project*> directory. The <*project*> name is to be specified by the user as a combination of 1-8 alpha-numeric characters. The <*project*> directory contains one or more <*case*>, <*code*>, and <*run*> directories, as shown in Figure 2. This classification enables to have unique identifications for computations of different physical cases by different numerical models with different run parameters. Note that there are different directory names but the same file names, thus the unique file identification requires to specify directory name as well.

2.2 Case

The `case.`<*case*> directory contains input data. The <*case*> name is derived from the user specified names into one of the following three possible structures:

<*initial values*>.<*grid*>

<*boundary values*>.<*grid*>

<*initial values*>.<*boundary values*>.<*grid*>

The <*initial values*> and <*boundary values*> are to be specified by the user as a combination of 1-8 alpha-numeric characters. The <*grid*> name is to be specified by the user with the syntax <*n1*>`x`<*n2*>`x`<*n3*>, where <*n1*>, <*n2*>, and <*n3*> are the number of cells in the each direction of the computational domain.

The `case.`<*case*> directory contains the following names: `ini.txt`, `bnd.txt`, `grd.nc`, `ini.nc`, `bnd.nc`, `ini.png`, and `bnd.png`. Files `ini.txt` and `bnd.txt` store names and values of parameters used in a creation of `ini.nc` `bnd.nc` files, respectively. Files `ini.png` and `bnd.png` are representative images showing the main characteristics of initial and boundary data, respectively.

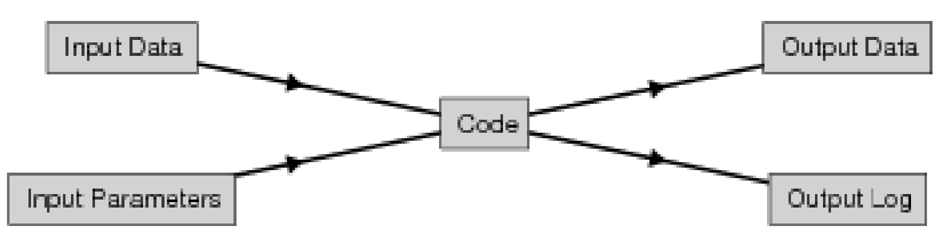

Figure 1: Basic computational scheme.

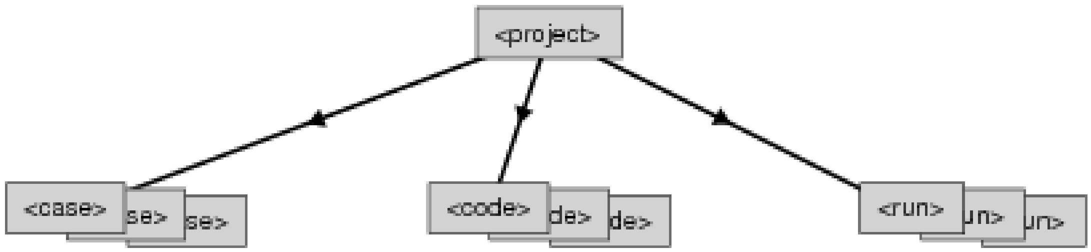

Figure 2: Project directory structure.

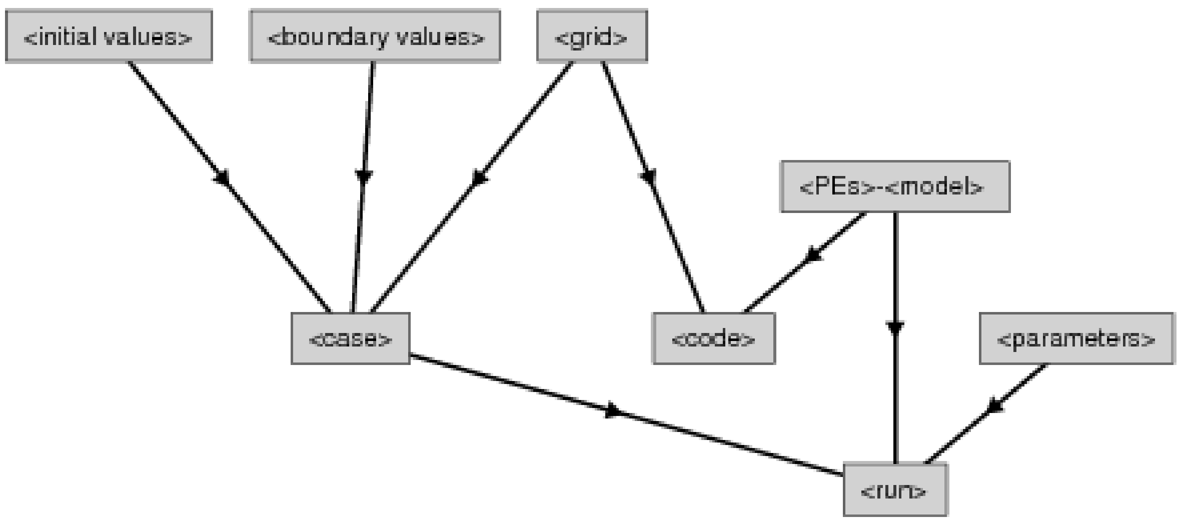

Figure 3: Hierarchy of naming conventions.

2.3 Code

The `code.`<*code*> directory contains array dimensions, makefile, and compiled executable. The <*code*> name is a derived name (see Figure 3) and it has the following naming structure:

<*grid*>.<*PEs*>-<*model*> ,

where <*grid*> is the number of grid cells (in the form of <*n1*>x<*n2*>x<*n3*>), <*PEs*> is the number of processing elements, and <*model*> is a combination of 1-8 alpha-numeric characters defining options selected during the pre-processing step of the compilation. All these three names are to be specified by the user. Note that <*grid*> specification for the <*code*> has to be identical with the specification for the <*case*> (see the previous Section).

The `code.`<*code*> directory contains the following names: `code.txt`, `Makefile`, `comdim.F`, `enlil`, `merim`. File `code.txt` store names and values of parameters that define the code options and array dimensions that were used to create of `Makefile` and `comdim.F` files. Files `Makefile` and `comdim.F` are used at code compilation and files `enlil` and `merim` are binary executables.

2.4 Run

The `run.`<*run*> directory contains input parameters, output data, output log, and, in the case of a parallel system, batch script and errors log. The <*run*> name has the following structure:

<*case*>.<*PEs*>-<*model*>.<*parameters*> ,

(see Figure 3 for specification of <*case*>). The <*parameters*> name is a combination of 1-8 alpha-numeric characters. Note: by using <*PEs*>-<*model*> instead of <*code*> there is only one <*grid*> name used in the <*run*> name, since the <*grid*> names are identical in <*case*> and <*code*> names.

The `run.`<*run*> directory contains the following names: `par.txt`, `in`, `out`, `run`, `job`, `tim.`*rrrr*`.nc`, `evh.nc`, `evl.nc`, `evg.nc`, and `evp.nc`. File `par.txt` store names and values of parameters that define were used to create `in`, `run`, and `job` files. File `out` contains messages produces during the computation and files with `.nc` suffix contain various output data.

3 Input Data

Input data are initial and/or boundary values used to "drive" the numerical computations. Input data are stored as NetCDF files and involve attributes, dimensions, and values of variables (stored in the metric SI system of physical units).

3.1 Files

There are four possible input data files as given in Table 1. At least one of the data file has to be provided by the user. If only initial values are provided (file type `ini` or `res`) then constant values (extrapolated from the computational domain at the beginning of computations) are used at the inner boundary. If only boundary values are provided (file type `bnd` or `pas`) then initial values at the beginning of computations are extrapolated from the boundary values and numerical relaxation is used to obtain physically consistent initial state.

The ENLIL code can produce up to eight data files given in Table ?? depending on values of input parameters provided for computations. In the file type `tim`, <*rrrr*> is a 4-digit identification of the time level (record number), with leading zeros. In case of parallel computations, the `tim.`*rrrr*`.nc` file is produced (after finishing the computation) by the `MERTIM` procedure that merges <*pppp*>`.tim.`<*rrrr*>`.nc` files produced by individual processing element on a parallel system (<*pppp*> is a 4-digit identifications of the processing element (slab number)with leading zeros). Meaning of the `x*s.nc` and `ev*.nc` files is illustrated in Figures 4 and 5, respectively.

Table 1: Input data files

File name	Precision	Description
`ini.nc`	Double	Initial values for the new computation
`res.nc`	Double	Initial values from the previous computation
`bnd.nc`	Double	Boundary values at the inner boundary
`pas.nc`	Double	Boundary values from the previous computation

3.2 Attributes

Attributes are descriptive labels associated with the each NetCDF file (global attributes) and variable (variable attributes) and they are listed in Tables 2 and 3, respectively.

3.3 Dimensions

Dimensions of the NetCDF dataset define the shape and size of arrays containing data for individual variables. Each spatial dimension has associated two array dimensions to distinguish between data stored at cell centers or cell interfaces. A naming conventions is used in which the individual spatial dimensions are numbered 1, 2, and 3, followed by a label `h` if data at cell interfaces (there are just numbers withou label `h` if data are at cell centers). Numbered dimensions 1, 2, and 3 corresponds to dimensions x, y, and z in Cartesian, r, ϕ, and z in cylindrical, and r, θ, and φ in spherical coordinate systems.

Different files contain arrays with different dimensions as given in Table 5. Their spatial dimensions have to correspond to the <*grid*> values specified at the compilation of the <*model*>.

Dimensions of the NetCDF dataset define the shape and size of numerical arrays containing values of individual variables. There are up to eight dimensions that can be used in input and output data files as given in Table 4.

Table 2: Attributes of input data files

Name	Description
`type`	File type (see Tab. 1)
`title`	File content label
`name`	Name of <*case*> (see Section 2.2)
`project`	Project name label
`initial`	Initial values label
`resume`	Resume values label
`boundary`	Boundary values label
`passage`	Passage values label
`grid`	Numerical grid label
`geometry`	Coordinate system (`cartesian`, `cylindrical`, or `spherical`)
`code`	Computational code label
`parameters`	Run parameters label
`refdate_mjd`	Reference date in Modified Julian days (=JD-2400000.5)
`history`	Date and time of the file origin

Table 3: Attributes of input data variables

Name	Description
`long_name`	A long descriptive name (title)
`units`	Physical unit used for the variable's data

Spatial dimensions (`n1, n2, n3, n1h, n2h, n3h`) define the numerical mesh. A naming convention is used in which the individual spatial dimensions are numbered "1", "2", and "3". Arrays with dimensions `n1`, `n2`, and `n3` (`n1h`, `n2h`, and `n3h`) contain data at cell centers (cell interfaces) of the numerical mesh. Additional dimensions `nblk` and `ntime` are used in case of the AMR computations and storing data at different time levels, respectively.

Different array dimensions are used for main variables in the computational domain (Table ??), at the inner boundary (Table ??), and stored at the given positions (Table ??).

Table 4: Dimensions of variables

Name	Array dimensions
`n1`	Number of cells in X1-direction
`n2`	Number of cells in X2-direction
`n3`	Number of cells in X3-direction
`n1h`	Number of cell interfaces in X1-direction
`n2h`	Number of cell interfaces in X3-direction
`n3h`	Number of cell interfaces in X3-direction
`nblk`	Number of computational blocks
`ntime`	Number of time levels

Table 5: Dimensions of input data variables

File name	Array dimensions
`ini.nc`	$n1 \times n2 \times n3 \times nblk$
`res.nc`	$n1 \times n2 \times n3 \times nblk$
`bnd.nc`	$1 \times n2 \times n3$
`pas.nc`	$2 \times n2 \times n2$

3.4 Variables

Input and output files have three main data structures as given in Tables ??, ??, and ??. Variables are spatial coordinates (`x1`, `x2`, `x3`, `x1h`, `x2h`, and `x3h`), physical time and numerical timestep (`time`, `dtstep`), parameters (`gamma`), and main variables (`d, t, v1, v2, v3, b1, b2, b3, dp, bp`). Numbers 1, 2, and 3 correspond to components x, y, and z in Cartesian, r, ϕ, and z in cylindrical, and r, θ, and φ in spherical coordinate systems. Note that variables `dp` and/or `bp` are used in certain applications only.

Input and outputs data files that contain spatial distribution of values at given time levels (file types `res`, `tim`, `x1s`, `x2s`, and `x3s`) involve variables given in Table ??.

Boundary data files contain time-dependent values at the inner boundary of the computational domain. These files (`bnd.nc` and `pas.nc`) involve variables given in Table ?? where *nb1* is 1 or 2 for `bnd.nc` or `pas.nc` file,

Temporal evolution files (`evh.nc`, `evl.nc`, `evg.nc`, and `evp.nc`) contain variables given in Table ?? where *nobs* is `nhel`, `nsel`, `ngeo (=5)`, or `npla (<=6)` for `evh.nc`, `evl.nc`, `evg.nc`, or `evh.nc` file, respectively.

3.5 Variables

Input data files contain data given in Table 6

Table 6: Variables of input data

Name	Description	Units
`x1(n1)`	Heliospheric position of cell centers	m
`x2(n2)`	Meridional position of cell centers	rad
`x3(n3)`	Azimuthal position of cell centers	rad
`x1h(n1h)`	Heliospheric position of cell interfaces	m
`x2h(n2h)`	Meridional position of cell interfaces	rad
`x3h(n3h)`	Azimuthal position of cell interfaces	rad
`time`	Physical time	s
`dtstep`	Time step	s
`gamma`	Ratio of specific heats	–
`d(n1,n2,n3)`	Mass density	kg/m^3
`t(n1,n2,n3)`	Temperature	K
`v1(n1,n2,n3)`	Radial velocity	m/s
`v2(n1,n2,n3)`	Meridional velocity	m/s
`v3(n1,n2,n3)`	Azimuthal velocity	m/s
`b1(n1,n2,n3)`	Radial magnetic field	T
`b2(n1,n2,n3)`	Meridional magnetic field	T
`b3(n1,n2,n3)`	Radial magnetic field	T
`dp(n1,n2,n3)`	Cloud mass density	kg/m^3
`bp(n1,n2,n3)`	Magnetic field polarity	–

Numbered dimensions 1, 2, and 3 corresponds to dimensions x, y, and z in Cartesian, r, ϕ, and z in cylindrical, and r, θ, and φ in spherical coordinate systems.

4 Output Data

Output data are stored as NetCDF files and involve attributes, dimensions, and values of variables (stored in the metric SI system of physical units).

4.1 Files

The ENLIL code can produce data files given in Table 7 depending on values in input parameters provided for computations.

Table 7: Output data files

Name	Precision	Description
`res.nc`	Double	Restart values for resuming the previous computation
`tim.`*rrrr*`.nc`	Single	Values within the computational domain at given time levels
`x1s.nc`	Single	Values on the Theta-Phi slice at R and at given time levels
`x2s.nc`	Single	Values on the R-Phi slice at Theta and at given time levels
`x3s.nc`	Single	Values on the R-Theta slice at Phi and at given time levels
`evh.nc`	Single	Temporal evolution at given positions in heliosphere
`evl.nc`	Single	Temporal evolution at given positions on Sun-Earth line
`evg.nc`	Single	Temporal evolution at geospace positions
`evp.nc`	Single	Temporal evolution at planetary positions

Note that the `tim` file is produced (after the computation is finished) by the `MERTIM` procedure that merges `tim` files

<*pppp*>`.tim.`<*rrrr*>`.nc`,

produced by individual processing element on a parallel system (<*pppp*> and <*rrrr*> are 4-digit identifications of the processing element (slab number) and time level (record number), with leading zeros, respectively).

4.2 Attributes

Attributes are descriptive labels associated with the each NetCDF file (global attributes) and variable (variable attributes) and they are listed in Tables 8 and 9, respectively. Note that global attributes are replicated from input data, code specification label, and the very first line of input parameters.

Table 8: Attributes of output data files

Name	Description
`type`	File type (see Tab. 7)
`title`	File content label
`name`	Name of <*run*> (see Section 2.4)
`project`	Project name label
`initial`	Initial values label
`resume`	Resume values label
`boundary`	Boundary values label
`passage`	Passage values label
`grid`	Numerical grid label
`geometry`	Coordinate system (`cartesian`, `cylindrical`, or `spherical`)
`code`	Computational code label
`parameters`	Run parameters label
`refdate_mjd`	Reference date in Modified Julian days (=JD-2400000.5)
`history`	Date and time of file origin

Table 9: Attributes of output data variables

Name	Description
`long_name`	A long descriptive name (title)
`units`	Physical unit used for the variable's data

4.3 Dimensions

Dimensions for a NetCDF dataset defines shape and size of arrays containing data for individual variables. There are two main groups of NetCDF files.

Files with values at given time levels (file types: `ini`, `res`, `bnd`, `pas`, `tim`, `x1s`, `x2s`, and `x3s`) use up to seven dimensions to specify arrays depending on the file type and the particular variable. As in Section 3, the individual spatial dimensions are numbered 1, 2, and 3, and values located at the cell centers unless a label `h` is used.. Numbered dimensions 1, 2, and 3 corresponds to dimensions x, y, and z in Cartesian, r, ϕ, and z in cylindrical, and r, θ, and φ in spherical coordinate systems. Finally, the temporal dimension is used in file types `bnd`, `pas`, `x1s`, `x2s`, and `x3s`) and it corresponds to the number of time levels at which data are stored.

Different files contain arrays with different dimensions as given in Table 10. Files with temporal evolution at given points (file types: `evh`, `evl`, `evg`, and `evp`) have arrays with the first dimension corresponding to the number of observing points, and with the second dimension corresponding to the number of time levels.

Table 10: Dimensions of output data variables

File name	Array dimensions
`res.nc`	n1 × n2 × n3 × nblk
`tim.`*rrrr*`.nc`	n1 × n2 × n3 × nblk
`x1s.nc`	1 × n2 × n3
`x2s.nc`	n1 × 1 × n3
`x3s.nc`	n1 × n2 × 1
`evh.nc`	nhel × ntime
`evl.nc`	nsel × ntime
`evg.nc`	ngeo × ntime
`evp.nc`	npla × ntime

Table 11: Variables of spatial-distribution data

Name	Description	Units
`x1(n1)`	Heliospheric position of cell centers	m
`x2(n2)`	Meridional position of cell centers	rad
`x3(n3)`	Azimuthal position of cell centers	rad
`x1h(n1h)`	Heliospheric position of cell interfaces	m
`x2h(n2h)`	Meridional position of cell interfaces	rad
`x3h(n3h)`	Azimuthal position of cell interfaces	rad
`time`	Physical time	s
`dtstep`	Time step	s
`gamma`	Ratio of specific heats	–
`d(n1,n2,n3)`	Mass density	kg/m^3
`t(n1,n2,n3)`	Temperature	K
`v1(n1,n2,n3)`	Radial velocity	m/s
`v2(n1,n2,n3)`	Meridional velocity	m/s
`v3(n1,n2,n3)`	Azimuthal velocity	m/s
`b1(n1,n2,n3)`	Radial magnetic field	T
`b2(n1,n2,n3)`	Meridional magnetic field	T
`b3(n1,n2,n3)`	Radial magnetic field	T
`dp(n1,n2,n3)`	Cloud mass density	kg/m^3
`bp(n1,n2,n3)`	Magnetic field polarity	–

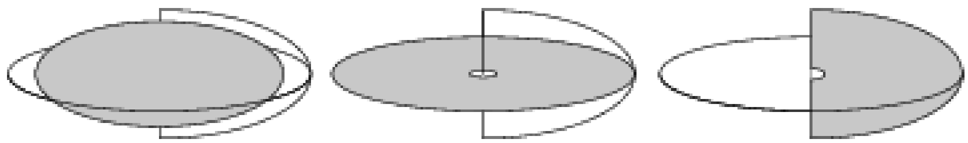

Figure 4: Output files with 2-D slices.

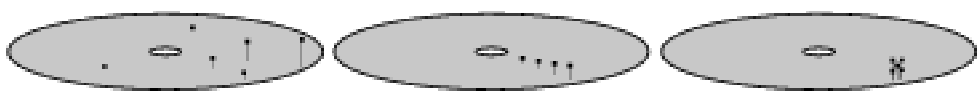

Figure 5: Output files with temporal evolution at points.

Table 12: Variables of temporal evolution data

Name	Description	Units
`x1(`*nobs*`,ntime)`	Heliospheric position	m
`x2(`*nobs*`,ntime)`	Meridional position	rad
`x3(`*nobs*`,ntime)`	Azimuthal position	rad
`time(ntime)`	Physical time	s
`dtstep(ntime)`	Time step	s
`gamma(ntime)`	Ratio of specific heats	–
`d(`*nobs*`,ntime)`	Mass density	kg/m^3
`t(`*nobs*`,ntime)`	Temperature	K
`v1(`*nobs*`,ntime)`	Radial velocity	m/s
`v2(`*nobs*`,ntime)`	Meridional velocity	m/s
`v3(`*nobs*`,ntime)`	Azimuthal velocity	m/s
`b1(`*nobs*`,ntime)`	Radial magnetic field	T
`b2(`*nobs*`,ntime)`	Meridional magnetic field	T
`b3(`*nobs*`,ntime)`	Radial magnetic field	T
`dp(`*nobs*`,ntime)`	Cloud mass density	kg/m^3
`bp(`*nobs*`,ntime)`	Magnetic field polarity	–

4.4 Variables

There are two types of output data files and they contain different variables.

- Spatial distribution files (`res.nc`, `tim.nc`, `x1s.nc`, `x2s.nc`, and `x3s.nc`) contain variables given in Table 11

- Temporal evolution files (`evh.nc`, `evl.nc`, `evg.nc`, and `evp.nc`) contain variables given in Table 12, with values in single precision, where *nobs* is `nhel`, `nsel`, `ngeo`, or `npla` for `evh.nc`, `evl.nc`, `evg.nc`, and `evp.nc` file, respectively.

4.5 Mass Storage System (MSS)

NCAR's Mass Storage System (MSS) is a central, large-scale data archive that stores data used and generated by programs executed on NCAR's compute servers. The MSS is used from intermediate and long-term storage of files.

Files are stored in one of three locations, listed here in a hierarchy with the shortest access time firsts

1. the disk farm;
2. the robotic tape library
3. the offline tape archive

The MSS migrates files among these three storage systems to provide the most efficient use of the system. Where the files reside as a given time depends on their size and how frequently the files are requested.

5 Community Data Portal (CDP)

NCAR/SCD has established the Community Data Portal (CDP), a collection of earth science datasets from NCAR, UCAR, UOP, and participating organizations in the areas of oceanic, atmospheric, space weather, and turbulence research (`http://dataportal.ucar.edu`).

Results from selected numerical simulations are archived as data sets at the NCAR Supercomputer Center in Boulder, CO. These data sets are available via the Community Data Portal (CDP) that provides an interactive access to browsing, pre-viewing, and downloading of data sets. The CDP can be accessed at `http://dataportal.ucar.edu:8443/` where the user is asked for registration.

5.1 Classification

Results from heliospheric simulations are classified according to:

- Dimensionality:
  - spatial distribution at given time levels (2-D or 3-D arrays)
  - temporal evolution at given locations (in heliosphere, on Sun-Earth line, at geospace, at planetary positions)
  - special (magnetic field line)
- Computational code:
  - physical model
  - method of solution
  - numerical resolution
- Driver (time-dependent boundary values)
  - analytic models
  - empirical models
  - numerical models
- Conditions
  - ambient solar wind
  - transient disturbances

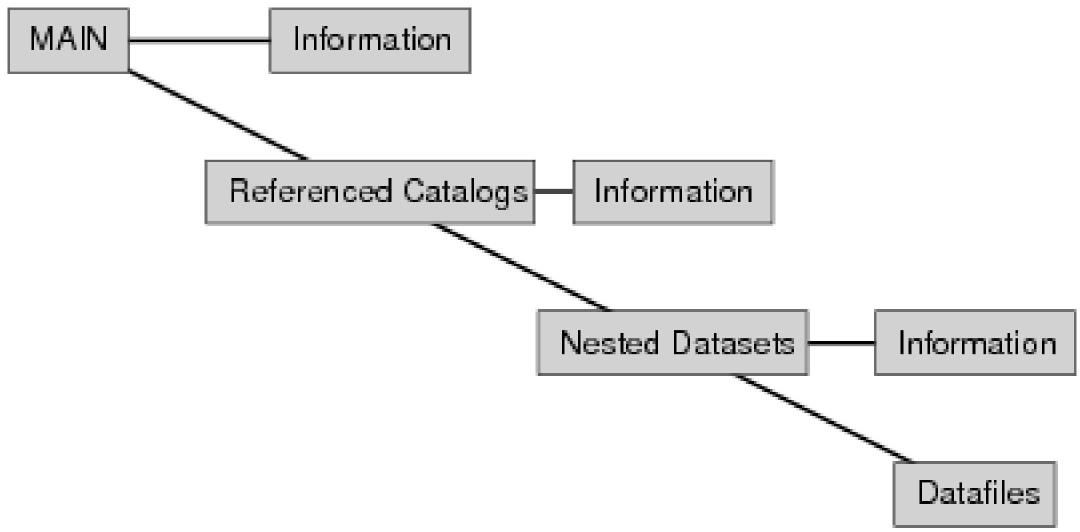

Figure 6: Main structure of the CDP data files.

- Events
  - hypothetic
  - observed

5.2 Main Structure

Data files accessible via the CDP are hierarchically organized as shown in Fig. 6.

5.3 Main Metadata

Main description of the available datasets is contained in the the main metadata according to a so-called Dublin Core Metadata as given in Table 13.

6 Incorporation of Data into the CDP

6.1 Main Metadata

To facilitate automatic conversion of heliospheric data into the CDP system, we are providing the `project.txt` file (Tab. 14) with information required by Dublin Core Metadata (Tab. 13).

6.2 Information and Data Files

Data archived at NCAR/MSS are then converted into NCAR/CDP using the relations given in Table 15.

6.3 Automatic Conversion

7 References

Table 13: Brief description of Dublin Core Metadata

Name	Description
`title`	Name given to the resource
`creator`	The person or organization primarily responsible
`subject`	Keywords or phrases describing the subject or content
`description`	A textual description of the content (abstract)
`publisher`	The entity responsible for making the resource available
`contributor`	A person or organization whose contribution is secondary
`date`	A date associated with the creation or availability of the resource
`type`	The category of the resource
`format`	The data format
`identifier`	A string used to uniquely identify the resource
`source`	Information about a second resource
`language`	`en`
`relation`	An identifier of a second resource
`coverage`	Spatial and/or temporal characteristics
`rights`	A rights management statement

Table 14: Example of the `project.txt` file

```
title 1997 May 12 Interplanetary Event
creator CU/CIRES and NOAA/SEC, Boulder, CO
creator SAIC, San Diego, CA
subject Space Weather
subject Solar Wind
subject Interplanetary Shock
description Boundary conditions, distribution and evolution
publisher NSF Center for Integrated Space Weather
contributor Dusan Odstrcil
date 2004
type collection
format NetCDF
identifier cu.cires.enlil
source Numerical code ENLIL
language en
relation http://sprg.ssl.berkeley.edu/cism
coverage 0.14-1.14 AU, 1997 May 12-22
rights freely available
```

Table 15: Relation between files

Files in MSS	Files in CDP	Comment
<project>	Catalog	
<project>/project.txt	Information	Dublin Core Metadata
<project>/<case>	Nested data set	
<project>/<case>/bnd.txt	Information	Boundary parameters
<project>/<case>/bnd.png	Information	Boundary image
<project>/<case>/grd.nc	NetCDF data	Numerical grid
<project>/<case>/bnd.nc	NetCDF data	Boundary conditions
<project>/<code>	Nested data set	
<project>/<code>/code.txt	Information	Code parameters
<project>/<code>/Makefile	–	
<project>/<code>/e4dim.F	–	
<project>/<code>/enlil	–	
<project>/<code>/mertim	–	
<project>/<run>	Nested data set	
<project>/<run>/par.txt	Information	Run parameters
<project>/<run>/tim.*rrrr*.nc	NetCDF data	Spatial distribution at given time levels
<project>/<run>/evh.nc	NetCDF data	Evolution at points in heliosphere
<project>/<run>/evl.nc	NetCDF data	Evolution at points on Sun-Earth line
<project>/<run>/evg.nc	NetCDF data	Evolution at geospace positions
<project>/<run>/evp.nc	NetCDF data	Evolution at planetary positions