GCMDAT Version 1.1
A Database of General Circulation Model Results

René Kernen & Dimitrios Gyalistras
Climatology and Meteorology Research Group
Institute of Geography
University of Bern

August 2002

Contents

Chapter 1

Introduction

Global climate change and its possible implications present an important branch of todays environmental research. Based on the observations of climate variables and the forcing agents that cause climate to change, the climate from the distant past to the present day is being reconstructed, using a variety of approaches. In order to provide quantitative estimates of possible future climate changes, numerical models, so called General Circulation Models (GCMs) of the Atmosphere and the ocean, have been developed (Albritton et al. 2001). GCMs are used to simulate the climate responses to different input scenarios of future forcing agents. The simulated results can be used to analyse the climate system or to assess likely impacts of climate change in a variety of sectors.

GCM simulations produce a huge amount of data. There have already been made specific efforts to make data available for climate analysis and impact studies, namely in the context of the Model Evaluation Consortium for Climate Assessment (MECCA), the Atmospheric and Coupled Model Intercomparison Projects (AMIP and CMIP) and last but not least the IPCC Data Distribution Center (IPCC DDC, http://ipcc-ddc.cru.uea.ac.uk/) which is a logical development of the publication of a set of climate scenarios and socio–economic projections for the IPCC WGII Assessment (Greco et al. 1994).

Although the IPCC DDC already means a substantial progress in providing climate data, it is still not suitable for operational analyses. First of all, this is because of the relatively slow, interactive user interface of the IPCC DDC web site. Research work needs batch processing and reasonably fast data access. Another important issue is the data format and quality. Moreover, although the GRIB (GRIdded Binary) format has been defined by the WMO as the standard for the exchange of climate data (WMO 1994), we found that the LoLa (Longitude – Latitude) format suits our needs better. The IPCC DDC provides data on an "as is basis", such that the data are provided on different grids and follow different storage conventions. Also, no quality checks are normally performed by the IPCC DDC which provides only raw data, directly as obtained from the modelling centres.

Therefore we decided to create our own database, that was designed to provide fast access to quality tested data through a simple and easy-to-use interactive and progamming user interface. This database was named GCMDAT. GCMDAT presently includes only data obtained from the IPCC DDC. Moreover, data provided directly by the various modelling centers could easily be added at a later stage.

The first part of this report describes the datasets, available in the newest version (1.1) of GCMDAT (chapter 2) as well as how to access them (3). The second part is related to the design and implementation of GCMDAT and the preparation (4.2) and quality testing (4.3) of the data.

Chapter 2

Overview of Available Data Sets in GCMDAT V1.1

The IPCC DDC (IPCC 1998) provides data for a wide range of GCMs. Table 2.1 shows an overview of the models for which data are available from the IPCC DDC (as of 30/08/02) or will be made available in the near future. Most of these models are also represented in GCMDAT V1.1.

Table 2.1: Overview of Models

No	Model Description	Resolution[1]	Reference
1	CCSR/NIES - Center for Climate System Research, University of Tokyo / National Institute for Environmental Studies	A: T21 L20 O: 2.8x2.8 L17	Emori et al. (1999)
2	CGCM1 - Canadian Global Coupled Model 1	A: T32 L10 O: 1.8x1.8 L29	Flato et al. (2000), Boer et al. (2000a), Boer et al. (2000b)
3	CGCM2 - Canadian Global Coupled Model 2	A: T32 L10 O: 1.8x1.8 L29	Flato und Boer (2001)
4	CSIRO - Commonwealth Scientific & Industrial Research Organisation	A: R21 L9 O: R21L21	Gordon und O'Farrell (1997)
5	CSM - NCAR Community Climate Model	A: T42 L18 O: 2.0x2.4 L45	Boville und Gent (1998), Boville et al. (2001)
6	DOE PCM - DOE Parallel Climate Model	A: T42 L18 O: 0.66x0.66 L32	Washington et al. (2000)
7	ECHAM3/LSG - ECMWF Model, modified in HAMburg/Large Scale Geostrophic Ocean GCM	A: T21 L19 O: 4.0x4.0 L11	Cubasch et al. (1995)
8	ECHAM4/OPYC3 - ECMWF Model, modified in HAMburg/Isopycnal Ocean Model	A: T42 L19 O: 2.8x2.8 L11	Roeckner et al. (1999)
9	GFDL_R15 - Geophysical Fluid Dynamics Laboratory	A: R15 L9 O: 4.5x3.7 L12	Manabe et al. (1991), Manabe et al. (1992)
10	GFDL_R30 - Geophysical Fluid Dynamics Laboratory	A: R30 L14 O: R30 L18	Delworth et al. (2002), Dixon et al. (2002)
11	HadCM2 - Hadley Centre Climate Model 2	A: 2.5 x 3.75 L19 O: 2.5 x 3.75 L20	Johns et al. (1997)
12	HadCM3 - Hadley Centre Climate Model 3	A: 2.5 x 3.75 L19 O: 1.25 x 1.25 L20	Gordon et al. (2000)

Several forcing scenarios have been used to drive these models. A detailed discussion of the scenarios is given in the Special Report on Emission Scenarios (SRES) by the IPCC (Nakicenovic und Swart 2000). Table 2.2 shows an overview of all available scenarios and variables together with the future time periods covered by the individual simulated data sets in GCMDAT V1.1. All data currently included in GCMDAT are monthly means.

[1] Horizontal and vertical resolution for atmosphere (A) and ocean (O), TXX: Spectral Model, Triangular Truncation at wavenumber XX, RXX: Spectral Model, Rhomboidal Truncation at wavenumber XX, LYY: Number (YY) of vertical levels

Table 2.2: Overview of GCM data sets, made available by the IPCC DDC (as of 30/08/02). X denotes data sets included in GCMDAT V1.1

No	Model	Forcing Scenario[2]	Time Period	Datasets[3]			
				slp	tem	pre	gph
1	CCSR/NIES	CTRL	1890–2099	-	-	-	-
2		GHG1	1890–2099	X	X	X	-
3		GHG1A	1890–2099	X	X	X	-
4	CGCM1	CTRL	200 yrs	X	X	X	-
5		GHG1	1900–2100	X	X	X	-
6		GHG1A	1900–2100	X	X	X	-
7	CGCM2	CTRL	1890–2099	-	-	-	-
8		GHG1A	1900–2100	X	X	X	-
9		A2	1990–2100	X	X	X	X
10		B2	1990–2100	X	X	X	X
11	CSIRO	CTRL	220 yrs	-	-	-	-
12		GHG1	1881–2100	X	X	X	-
13		GHG1A	1881–2100	X	X	X	-
14		A2	1961–2100	X	X	X	-
15		B2	1961–2100	X	X	X	-
16	CSM	CTRL	220 yrs	-	-	-	-
17		GHG1	130 yrs	-	-	-	-
18		A1	1980–2100	-	-	-	-
19		A2	1980–2100	-	-	-	-
20		B2	1980–2100	-	-	-	-
21	DOE PCM	CTRL	300 yrs	-	-	-	-
22		A2	1980–2099	X	X	X	-
23		B2	2001–2099	X	X	X	-
24	ECHAM3/LSG	CTRL	several 100 yrs	-	-	-	-
25		GHG1	1860–2100	X	X	X	-
26		GHG1A_1	1860–2100	X	X	X	-
27		GHG1A_2	1860–2100	-	-	-	-
28	ECHAM4/OPYC3	CTRL	240 yrs	X	X	X	-
29		GHG1	1860–2099	X	X	X	-
30		GHG1A_1	1860–2049	X	X	X	-
31		GHG1A_2	1860–2049	X	X	X	-
32	GFDL_R15	CTRL	1000 yrs	-	-	-	-
33		GHG1	1958–2057	-	-	-	-
34		GHG1A	1766–2056	-	-	-	-
35	GFDL_R30	CTRL	900 yrs	X	X	X	-
36		GHG1	1958–2057	X	X	X	-
37		GHG1A_1	1958–2057	X	X	X	-
38		GHG1A_2	1766–2065	X	X	X	-
39	HadCM2	CTRL	1000 yrs	-	-	-	-
40		GHG1	1860–2099	X	X	X	-
41		GHG1	1860–2099	-	-	-	-
42		GHG05	1860–2099	-	-	-	-
43		GHG05	1860–2099	-	-	-	-
44	HadCM3	CTRL	1000 yrs	X	X	X	-
45		GHG1	1860–2099	X	X	X	-
46		GHG1A	1860–2099	X	X	X	-
47		A2	1860–2099	X	X	X	X
48		B2	1860–2099	X	X	X	X

[2] CTRL = control run (present day forcing),
GHG1 = simulation driven by changing greenhouse gas concentrations only (IS92a),
GHG1A = greenhouse gases + sulphate aerosols (IS92a),
A2 = greenhouse gases + sulphate aerosols (SRES_A2),
B2 = greenhouse gases + sulphate aerosols (SRES_B2)

[3] slp = sea level pressure,
tem = temperature,
pre = precipitation,
gph = geopotential height

Chapter 3

Using GCMDAT V1.1

3.1 Data Formats

All GCM data are stored in Longitude – Latitude (LoLa) format (Waszkewitz et al. 1996). They are compressed using gzip. The metadata in the administration directory are stored as text files. Data that are extracted with the aid of one of the provided user interfaces come in lola format.

3.2 Installation

GCMDAT runs on UNIX Workstations only and is installed in 3 Steps: (1) Installation of all required progammes, (2) provision of the data files and (3) configuration of the Operating System (OS).

Step 1 The GCMDAT V1.1 requires, that the following programmes are installed on the workstation:

- PINGO - Procedural INterface for GRIB formatted Objects (Waszkewitz et al. 1996).
- GFDBI - Geophysical Fields Data Base Interface (written in Modula 2)
- GFDB - Geophysical Fields Data Base (main script, Korn Shell)

Step 2 The data files of GCMDAT can reside on any network drive that is accessible to the workstation (e.g. */mount/hostname/GCMDAT_V1.1*). Read access to the following directories is needed:

- the GCMDAT main directory (e.g. */mount/hostname/GCMDAT_V1.1*)
- the directories containing the model data (e.g. */mount/hostname/GCMDAT_V1.1/CCC1*)
- the database administration directory (e.g. */mount/hostname/GCMDAT_V1.1/_GFDB*), which should contain 3 parameter files (GFDB.DataLocations, GFDB.DataSources, GFDB.Datasets) that contain all needed meta information.

Step 3 An environment variable, called *GFDBDIR* must be defined that contains the name of the database administration directory (e.g. *GFDBDIR=/mount/hostname/GCMDAT_V1.1/_GFDB*).

```
alp$ GFDB -i

 Welcome to GFDB.


               ==================================================
                  Main menu
               ==================================================
                  1  Show data sources
                  2  Show datasets
                  3  Show attributes lists
                  4  Define data to extract
                ( 5  Extract data )
               --------------------------------------------------
                  q  Quit program
               ==================================================

 Your choice:
```

Figure 3.1: The main menu of the GCMDAT interactive user interface

```
alp$ GFDB -b . TrenbMSLP monthly mean r72x37 slp 1013 1899 2001 0.0 355.0 15.0
85.0


 Searching data base...
 Extracting data...

 Written:
"./TrenbMSLP.r72x37.0.0_355.0.15.0_85.0.monthly.mean.slp.1013.1899.2001.lola"
```

Figure 3.2: Example for the extraction of a data set using the GCMDAT programming interface

3.3 Data Extraction

Basically there are three ways of using GCMDAT: via the UNIX OS by accessing directly the model directories, through a simple, interactive user interface or by means of the programmers interface which allows for easy batch processing.

The access via the UNIX OS is straightforward since the naming of the directories and individual files should be self–explanatory (see also next section). However, most users will probably prefer to access the data via one of the two available, easy–to–use interfaces, which have been specifically designed to support the handling of large geophysical data sets.

The interactive user interface is based on a hierarchy of menus, which are displayed in text format in the terminal window (Fig. 3.1) and can be started by typing *GFDB –i* on the command line. It's use ist quite straightforward and needs no further explanation. Only one dataset at a time can be extracted.

In operational analysis, very often batch processing is used. This means that a program needs to extract data from the GCMDAT without any user interaction. The appropriate command is *GFDB –b* followed by a series of query parameters (Fig. 3.2). Further information can be found by typing the command *GFDB* without any arguments.

Chapter 4

Technical Description of the Database

4.1 Design and Implementation

4.1.1 The ”Geophysical Fields Data Base” (GFDB) System

GCMDAT presents a specific implementation of a ”Geophysical Fields Data Base” (GFDB). The GFDB data model was developped and implemented for Sun workstations by D. Gyalistras. Below we outline the general features of any data base that follows the GFDB system, including GCMDAT.

4.1.1.1 Specification

The development of GFDB was based on the following considerations:

- Simple and flexible structure of data base.
- Simple data formats.
- Handling of large amounts of data (many Gigabytes).
- Support of distributed data storage.
- Minimum labour effort for setting up, maintenance and updating.
- Maximum degree of self-documentation.
- Flexible support of multiple users.
- Efficient and flexible data access.
- Portability.
- Low cost.

4.1.1.2 Design

The basic entities in a GFDB are *DataSource*, *DataSet*, and *DataLocation*. A DataSource is used to summarize a family of DataSets under a common name. The following relatioships hold:

$$\boxed{\begin{aligned} DataSource : DataLocation &= 1 : 1 \\ DataSource : DataSet &= 1 : N \end{aligned}}$$

The entities are defined as follows:

DataSource = (dataSourceNr, dataSourceId, dataSourceDescription)

dataSourceNr = INTEGER.

dataSourceId = IDENTIFIER.

dataSourceDescription = STRING.

DataLocation = (dataSourceId, physicalLocation, directoryStructure)

dataSourceId = IDENTIFIER.

dphysicalLocation = STRING.

directoryStructure = ”flat” | ”hierarchical”.

DataSet = (dataSourceId, temporalResolution, statistic, variable, level, firstYear, lastYear, gridType, minLon, maxLon, minLat, maxLat, docuFileName)

dataSourceId = IDENTIFIER.

temporalResolution = ”hourly” | ”twoHourly” | ”fourHourly” | ”sixHourly” | ”twelveHourly” | ”daily” | ”monthly” | ”seasonal” | ”yearly”.

statistic = ”state” | ”mean” | ”stdDev” | ”min” | ”max” | ”median” | ”skewness,” | ”occurL” | ”occurH” | ”probL” | ”probH”.

variable = ”press” | ”slp” | ”presstend” | ”gp” | ”gph” | ”geoh” | ”temp” | ”vtemp” | ”ptemp” | ”paptemp” | ”maxtemp” | ”mintemp” | ”tempanom” | ”pressanom” | ”gphanom” | ”dd” | ”ff” | ”uwnd” | ”vwnd” | ”sf” | ”vpot” | ”msf” | ”scvv” | ”omega” | ”geovv” | ”vort” | ”div” | ”relvort” | ”reldiv” | ”ushear” | ”vshear” | ”curdirect” | ”curspeed” | ”ucur” | ”vcur” | ”shum” | ”rhum” | ”mixrat” | ”pwater” | ”vappress” | ”satdefic” | ”evap” | ”cloudice” | ”preciprate” | ”thundprob” | ”precip” | ”lprecip” | ”cprecip” | ”lsmask” | ”lrad” | ”srad” | ”globrad”.

level = INTEGER.

firstYear = INTEGER.

lastYear = INTEGER.

gridType = ”r72x36” | ”r72x37” | ”r180x90” | ”r180x91” | ”r360x180” | ”r360x181” | ”r720x360” | ”r720x361” | ” r96x73” | ”r144x73” | ”t21” | ”t30” | ”t42” | ”t106”.

minLon = REAL.

maxLon = REAL.

minLat = REAL.

maxLat = REAL.

docuFileName = STRING.

4.1.1.3 Implementation

Each entity is described by one data table which is stored in a text formatted file. The three text files must be named "GFDB.DataLocations", "GFDB.DataSources", and "GFDB.Datasets", respectively. They contain all meta-information needed to run a GFDB. They must be stored in a common directory, which is typically named "_GFDB".

All data belonging to a given data set are stored together in one directory as a series of gzipped, LOLA-formatted files. Each individual file contains all available data for one particular year. The file names fully describe the files' contents and are unique within a GFDB. The file names are constructed as follows:

4.1.2 Identifiers and Naming Conventions

Basically, every model has its own directory within the hierarchical file structure of the GCMDAT. Within this model directory, data are stored in different subdirectories, according to the current state of the work flow (e.g. downloaded data are stored in the _ORIG directory). The filenames are composed of a set of identifiers:

<filename> = <dataSourceId>.<temp_res>.<stat>.<grid>.<var>.<level>.<year>.<format>

Example: CCC2_GHG1A_M3.mon.mean.r72x36.temp.1013.2001.2050.lola.gz

<**dataSourceId**> identifier of data source

<**temp_res**> temporal resolution (e.g. mon = monthly)

<**stat**> statistical method (e.g. mean)

<**grid**> grid definition (e.g. r72x36 = regular grid with 72 longitudes and 36 latitudes)

<**var**> variable name (e.g. temp = temperature)

<**level**> vertical level (e.g. 1013 = sea level)

<**year**> time period, can be one (e.g. 2025) or more years (e.g. 2001.2050)

<**format**> data format (e.g. lola), can be compressed (e.g. lola.gz)

The name of the directory that contains the data files depends on the attribute *directoryStructure*. If *directoryStructure* = flat, all files must be located in a directory named according to the value specified for *physicalLocation*. If *directoryStructure* = hierarchical, the files must be placed in a directory named "*physicalLocation/temporalResolution.statistic.gridType.variable.level*".

Data access is accomplished directly via the UNIX/OS, or by means of an interactive or a programmer's interface. See chapter 3.

4.1.3 Specific Implementation of GCMDAT

GCMDAT was implemented using for all data sets *directoryStructure* = flat. The specific configuration of the GCMDAT Development Environment is shown in Figure 4.1. "_ADMIN", "_DOCU", "_ORIG" and "_PROGS" were additional directories which are not required by GFDB.

The dataSourceId was constructed as follows:

<**scen_id**> identifies a specific model run with a well defined forcing scenario

<scen_id> = <model_id>_<forcing_id>_<realization_nr_or_ensemble_mean_id>

<**model_id**> model (e.g. CCC2 = Canadian Center for Climate Modelling 2)

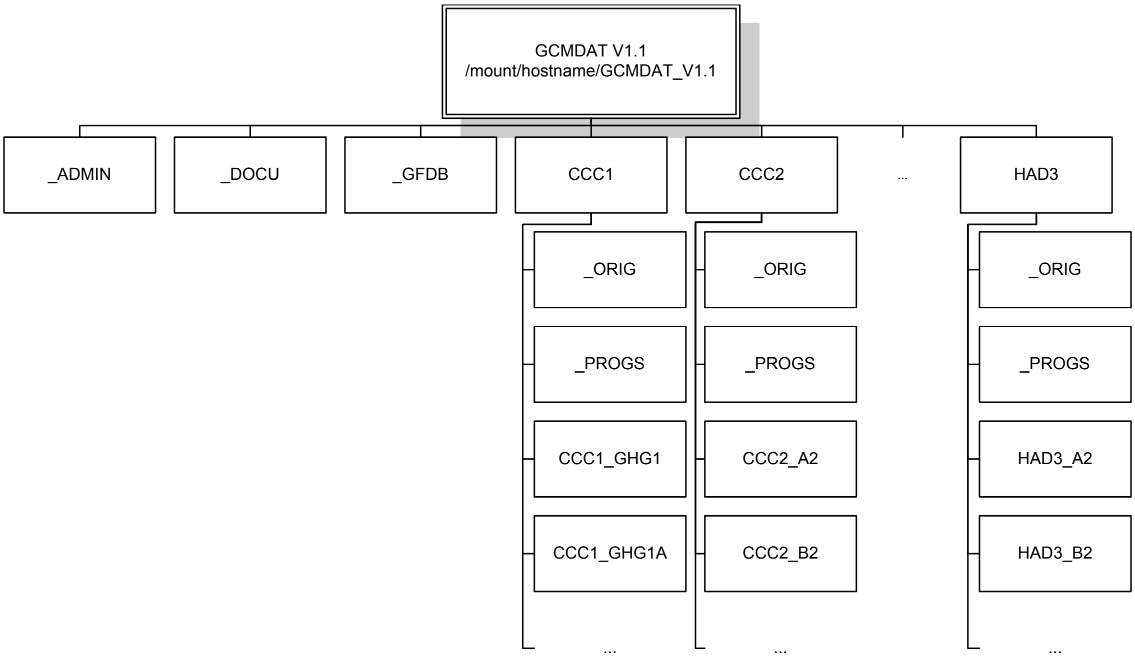

Figure 4.1: Hierarchical structure of GCMDAT within the UNIX filesystem

<**forcing_id**> forcing scenario (e.g. GHG1A = Greenhouse Gas Forcing with sulphate aerosols +1% CO_2)

<**realization_nr_or_ensemble_mean_id**> specific run, where several runs or an ensemble were calculated (e.g. M3 = mean of 3 ensemble runs)

4.2 Data Preparation

This section provides an overview of the data flow during preparation of a data set and the involved procedures. A graphical Overview is given in 4.2. The preparation of GCMDAT V1.1 consisted of two major steps. First, the insertion of the model data into the Database Development Environment. Secondly, the production of a release of GCMDAT. The Development Environment as well as GCMDAT are constructed according to the GFDB data model, which has been implemented on UNIX WS. Most used procedures were automated using korn shell scripts, which are also described in this section.

4.2.1 Korn Shell Scripts

4.2.1.1 Downloading of files and data and format conversion

Although some operations need still to be done manually, most of them have been automated. The Korn Shell provides a huge range of scripting possibility and the following scripts were of great use to ease and accelerate the whole data procurement process.

GetData Argument: parameter file.

Purpose: Downloads data from the IPCC DDC with ftp and saves them in the corresponding model directory (e.g. */mnt/hostname/GCMDAT_V1.1/CCC2/_ORIG*). Before calling this

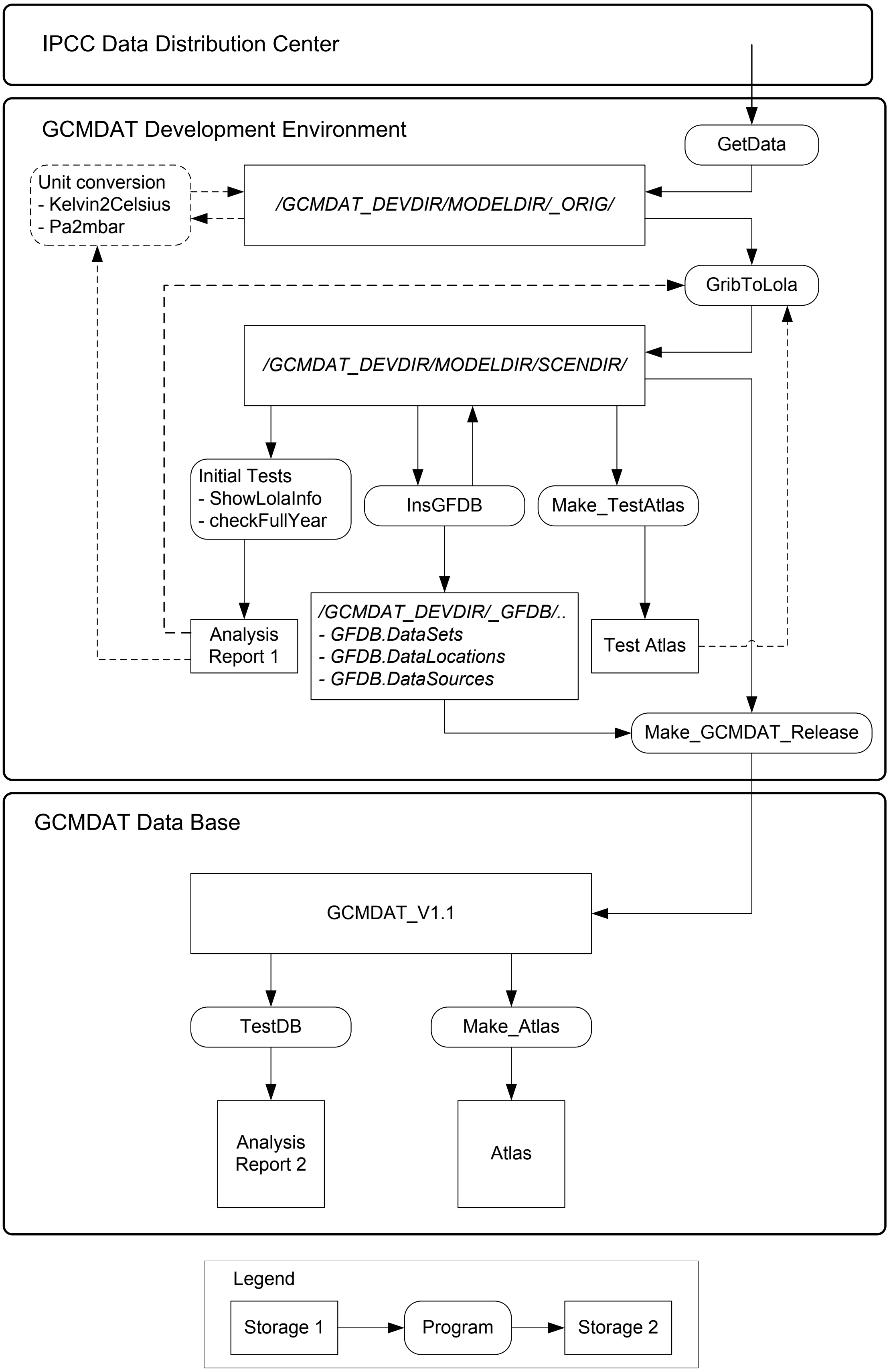

Figure 4.2: Overview of procedures used to develop GCMDAT V1.1

script, datasets have to be ordered at the IPCC DDC, and the parameters (e.g. file name, variable name) have to be written into the parameter file. This Script also calls *GribToLola* (see below).

GribToLola Arguments: prefix, GCMDAT directory, model ID, scenario ID, spatial resolution.

Purpose: Converts data from GRIB to LoLa, interpolates from irregular to regular grid and compresses files with gzip. Note that depending on the initial tests (see below), some files may need to be swapped!

4.2.1.2 Initial Tests

ShowLolaInfo Arguments: variable, year

Purpose: Produces an analysis report for a given variable and year, which can then be used to check physical units and minimum and maximum values for plausability.

- If temperature is in Kelvin, use *Kelvin2Celsius* to convert to Degree Celsius.
- Precipitation should be in millimeter per day.
- if Pressure is in Pa, use *Pa2mbar* to convert to mbar (or hPa).

checkFullYear Checks, if for all years complete data sets are provided including all 12 months!

If these initial tests are succesfull, data are ready for insertion into the GFDB of the Development Environment. When errors are encountered, downloaded data have to be corrected manually and GribToLola has to be rerun.

4.2.1.3 Insert data into GFDB

Inserts data into GFDB by splitting data into annual files and updating the corresponding metafiles.

split2annual split data files into annual files and save them.

updateDatasets update GFDB.Datasets

updateDataLocations update GFDB.DataLocations

updateDataSources update GFDB.DataSources

4.2.1.4 Visual Testing

Make_TestAtlas Argument: parameter file

Purpose: Creates a test atlas which enables an visual check of every dataset. If grid orientation is inversed (e.g. North and South swapped), invoke swaptopbottom with GribToLola.

4.2.1.5 Releasing of GCMDAT

Make_GCMDAT_Release Move data and metafiles into a newly created directory (GCMDAT_V1.0)

TestDB Tests database by simply reading each line in the GFDB.Datasets metafile, forming a query and extracting the corresponding data.

4.3 Quality Testing

Quality testing was done in two steps: The first step consisted of the above described initial tests and the production of a TestAtlas. A non-automated overall test was done, by comparing the TestAtlas with several public available plots from the different data centers, with the main focus on the pressure patterns (e.g. Azores High and Icelandic Low). Step two (TestDB) shows whether all data can be correctly extracted.

4.4 Maintenance

4.4.1 How to add a new dataset

- download data from IPCC DDC and save them to the corresponding model directory, using *GetData* and *GribToLola*.
- Run initial tests and do corrections, if necessary
- Insert data to the GFDB, using *InsGFDB*.
- Make a test atlas for visual checking of data
- If everything is OK so far, make a new release of GCMDAT, using *make_GCMDAT_Release*.
- With *TestDB* correct extraction of data is tested.
- Finally, create a new atlas, using *Make_Atlas*.

Glossary

A

AMIP Atmospheric Model Intercomparison Project

C

CMIP Coupled Model Intercomparison Project

G

GCM General Circulation Model

GFDB Geophysical Fields Data Base

GRIB GRidded BInary data format

I

IPCC Intergovernmental Panel on Climate Change

IPCC DDC IPCC Data Distribution Center ,

L

LoLa LOngitude–LAtitude format

M

MECCA Model Evaluation Consortium for Climate Assessment

O

OS Operation System

P

PINGO Procedural INterface for GRIB formatted Objects

S

SRES Special Report on Emission Scenarios

W

WMO World Meteorological Organisation

References

Albritton, D. L., L. G. Meira Filho, U. Cubasch, X. Dai, Y. Ding, D. J. Griggs, B. Hewitson, J. T. Houghton, I. Isaksen, T. Karl, M. M. V. P. Meleshko, J. Mitchell, M. Noguer, B. S. Nyenzi, M. Oppenheimer, J. E. Penner, S. Pollonais, T. Stocker, und K. E. Trenberth (2001). *Technical Summary.* In Houghton et al. (ed.) Climate Change 2001: The Scientific Basis. Contribution of Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change, S. 21–84.

Boer, G., G. Flato, und D. Ramsden (2000b). A transient climate change simulation with greenhouse gas and aerosol forcing: projected climate for the 21st century. *Climate Dynamics* **16**, 427–450.

Boer, G., G. Flato, M. Reader, und D. Ramsden (2000a). A transient climate change simulation with greenhouse gas and aerosol forcing: experimental design and comparison with the instrumental record for the 20th century. *Climate Dynamics* **16**, 405–425.

Boville, B. und P. Gent (1998). The NCAR Climate System Model and Version One. *Journal of Climate* **11**, 1115–1130.

Boville, B., J. Kiehl, P. Rasch, und F. Bryan (2001). Improvements to the NCAR CSM-1 for transient climate simulations. *Journal of Climate* **14**, 164–179.

Cubasch, U., G. A. Meehl, G. J. Boer, R. J. Stouffer, M. Dix, A. Noda, C. A. Senior, S. Raper, und K. S. Yap (2001). *Projections of Future Climate Change.* In Houghton et al. (ed.) Climate Change 2001: The Scientific Basis. Contribution of Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change, S. 525–582.

Cubasch, U., J. Waszkewitz, G. Hegerl, und J. Perlwitz (1995). Regional climate changes as simulated in time–slice experiments. *Climatic Change* **31**, 273–304.

Delworth, T., R. Stouffer, K. Dixon, M. Spelman, T. Knutson, A. Broccoli, P. Kushner, und R. Wetherald (2002). Simulation of climate variability and change by the GFDL R30 coupled climate model. *Climate Dynamics*. in press.

Dixon, K., T. Delworth, T. Knutson, M. Spelman, und R. Stouffer (2002). A Comparison of Climate Change Simulations Produced By GFDL Numerical Models Having Different Spatial Resolutions. *Global and Planetary Change*.

Emori, S., T. Nozawa, A. Abe-Ouchi, A. Numaguti, M. Kimoto, und T. Nakajima (1999). Coupled ocean–atmosphere model experiments of future climate change with an explicit representation of sulfate aerosol scattering. *Journal of the Meteorological Society of Japan* **77** (6), 1299–1307.

Flato, G. und G. Boer (2001). Warming Asymmetry in Climate Change Simulations. **28**, 195–198.

Flato, G., G. Boer, W. Lee, N. McFarlane, D. Ramsden, M. Reader, und A. Weaver (2000). The Canadian Centre for Climate Modelling and Analysis Global Coupled Model and its Climate. *Climate Dynamics* **16**, 451–467.

Gordon, C., C. Cooper, C. Senior, H. Banks, J. Gregory, T. Johns, J. Mitchell, und R. Wood (2000). The simulation of SST and sea ice extents and ocean heat transports in a version of the Hadley Centre coupled model without flux adjustments. *Climate Dynamics* **16**, 147–168.

Gordon, H. und S. O'Farrell (1997). Transient climate change in the CSIRO coupled model with dynamic sea ice. *Monthly Weather Review* **125** (5), 875–907.

Greco, S., R. Moss, D. Viner, und R. Jenne (1994). Climatic Scenarios and Socio-Economic Projections for the IPCC WGII Assessment. *IPCC Technical Support Unit, Washington D.C.*, 12 pp.

Houghton, J. T., Y. Ding, D. J. Griggs, M. Noguer, P. J. van der Linden, X. Dai, K. Maskell, und C. A. Johnson (ed.) (2001). *Climate Change 2001: The Scientific Basis. Contribution of Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge (UK) and New York: Cambridge University Press. 881 pp.

IPCC (1998). The IPCC Data Distribution Centre. <http://ipcc-ddc.cru.uea.ac.uk/index.html>. (last visited: 5 August 2002).

Johns, T., R. Carnell, J. Crossley, J. Gregory, J. Mitchell, C. Senior, S. Tett, und R. Wood (1997). The second Hadley Centre coupled ocean–atmosphere GCM: model description and spinup and validation. *Climate Dynamics* **13**, 103–134.

Manabe, S., M. Spelman, und R. Stouffer (1992). Transient response of a coupled ocean–atmosphere model to gradual changes of atmospheric CO2. Part II: Seasonal response. *Journal of Climate* **5**, 105–126.

Manabe, S., R. Stouffer, M. Spelman, und K. Bryan (1991). Transient response of a coupled ocean–atmosphere model to gradual changes of atmospheric CO2. Part I: Annual mean response. *Journal of Climate* **4**, 785–818.

Nakicenovic, N. und R. Swart (ed.) (2000). *Special Report on Emissions Scenarios*. Cambridge (UK): Cambridge University Press. 612 pp.

Roeckner, E., L. Bengtsson, J. Feichter, J. Lelieveld, und H. Rodhe (1999). Transient climate change simulations with a coupled atmosphere–ocean GCM including the tropospheric sulfur cycle. *Journal of Climate* **12**, 3004–3032.

Washington, W., J. Weatherly, G. Meehl, A. S. Jr., T. Bettge, A. Craig, W. S. Jr., J. Arblaster, V. Wayland, R. James, und Y. Zhang (2000). Parallel climate model (PCM) control and transient simulations. *Climate Dynamics* **16**, 755–774.

Waszkewitz, J., P. Lenzen, und N. Gillet (1996, October). The PINGO package - Procedural INterface for Grib formatted Objects. Technical Report 11, Deutsches Klima Rechen Zentrum. see also <http://www.dkrz.de/forschung/reports/report11-1.1/pingo-1.html>.

WMO (1994). Technical Report No. 17: Guide to WMO Binary Code Form GRIB 1. <http://www.wmo.ch/web/www/WDM/Guides/Guide-binary.html>. (last visited: 2 September 2002).

Appendix A

Appendix

A.1 Program Listings

A.1.1 Downloading of Data from IPCC DDC

A.1.1.1 GetData

```
#!/usr/bin/ksh
# ----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#=========================================
#
#  Name    :  GetData.ksh
#
#  Version :  1.0
#
#  Purpose :  Get Data from DKRZ (Deutsches Klima Rechen zentrum)
#
#===============================================================
#  Implementation and Revisions:
#---------------------------------------------------------------
#
# Author        Date            Description
# ------        ----            -----------
# rk            17/08/01        First implementation
# rk            12/09/01        add FILELIST
# rk            18/09/01        adapted to new directory structure
# rk                            renamed (GetFromDkrz -> GetECM3)
# rk            21/09/01        comments and echos (print) added
# rk            18/10/01        param.txt added
# rk            22/10/01        seasMean und Plots hinzugefuegt.
# rk            02/11/01        seasMeanMaps commented out
# rk            09/11/01        new Directory Structure implemented
# rk            06/12/01        new Prefix definition, gzip added
#                               new arguments for function call (GribToLola)
#
#============================================================================
# get parameterfile from commandline
PARAM=$1

# get GCMDATDIR and MODEL_ID from parameterfile
GCMDATDIR=‘grep GCMDATDIR $PARAM| sed -e ’s/GCMDATDIR=//’‘
MODEL=‘grep MODEL $PARAM| sed -e ’s/MODEL=//’‘

# set base path variables (MODELDIR,ORIGDIR,PROGDIR)
MODELDIR="$GCMDATDIR/$MODEL"
ORIGDIR="$MODELDIR/_ORIG"
PROGDIR="$MODELDIR/_PROGS"

   # does origdir exist?
   if [ ! -d "$ORIGDIR" ]
      then
      mkdir $ORIGDIR
   fi

# get NumExp from param.txt
NUMEXP=‘grep NumExp $PARAM| sed -e ’s/NumExp=//’‘

let COUNT=1
# for every line in $PARAM...
while [ "$COUNT" -le "$NUMEXP" ]; do
   set -A PARAMETERS ‘grep File$COUNT $PARAM‘
      DATAFILE=${PARAMETERS[1]}
      VARNAME=${PARAMETERS[2]}
      VARNAME=$(print $VARNAME | tr [A-Z] [a-z])
      SCENARIO=${PARAMETERS[3]}
      SPATIAL_RES="r72x36"                        #initial value
      TEMPORAL_RES="monthly"                      #initial value
      STATISTIC="mean"
      LEVEL="1013"

   # create filename prefix (HAD3_A2.monthly.mean.r72x36.temp.1013)
   SCEN=$MODEL"_"$SCENARIO
   PREFIX=$SCEN"."$TEMPORAL_RES.$STATISTIC.$SPATIAL_RES.$VARNAME.$LEVEL

   # ftp-transfer
   print "starting ftp transfer of \"$DATAFILE\"..."
   ftp ftp.dkrz.de <<EndFtp
```

```
   bi
   cd pub/cera
   get $DATAFILE $ORIGDIR/$PREFIX.grb
   bye
EndFtp
   print "... completed!"

   # Start Interpolation
   NEW_RESOLUTION="r72x36"
   $PROGDIR/GribToLoLa.ksh $PREFIX $GCMDATDIR $MODEL $SCEN $NEW_RESOLUTION
   gzip $ORIGDIR/$PREFIX.grb

   # next line
   let COUNT=COUNT+1
done
```

A.1.1.2 GribToLola

```
#!/usr/bin/ksh
# -------------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#===========================================
#
#  Name    :  GribToLoLa.ksh
#
#  Version :  1.0
#
#  Purpose :  Convert Data from GRIB (GRIdded Binary data format)
#             to LoLa (Longitude Latitude)
#
#=================================================================
#  Implementation and Revisions:  #
#-----------------------------------------------------------------
#
# Author Date Description
# --------------------
# rk 11/09/01 First implementation
# rk 22/10/01 filenames changed (included prefix)
# rk 09/11/01 new Directory Structure implemented
# rk 06/12/01 new arguments ,gzip added
#
#===================================================================
# set variables
PREFIX=$1
GCMDATDIR=$2
MODEL=$3
SCEN=$4
RESOLUTION=$5

# set base path variables (MODELDIR,ORIGDIR,SCENDIR)
MODELDIR="$GCMDATDIR/$MODEL"
ORIGDIR="$MODELDIR/_ORIG"
SCENDIR=$MODELDIR/$SCEN"."$RESOLUTION

   if [ ! -d "$MODELDIR" ]
      then
         mkdir $MODELDIR
   fi

   if [ ! -d "$SCENDIR" ]
      then
         mkdir $SCENDIR
   fi

   # Convert from GRIB to LoLa and
   # interpolate grid from irregular to regular
   print "Starting Interpolation of \"$PREFIX.grb\"..."
   lola interpolate $ORIGDIR/$PREFIX.grb $SCENDIR/$PREFIX.lola \
      < $GRIDS/$RESOLUTION.grid.asc
   gzip $SCENDIR/$PREFIX.lola
```

A.1.2 Initial Tests

A.1.2.1 ShowLolaInfo

```
#!/usr/bin/ksh
# -------------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#===============================================================================
#
#  Name    :  ShowLolaInfo.ksh
#
#  Version :  1.0
#
#  Purpose :
#
#===============================================================================
#  Implementation and Revisions:  #
#-------------------------------------------------------------------------------
#
# Author Date Description
# --------------------
# rk 13/11/01 First implementation
# rk 18/12/01 mknod implemented
#
#===============================================================================
VAR=$1
YEAR=$2
```

```
INFOFILE="lolainfo.$VAR.$YEAR.txt"

# set base path variables (GCMDATDIR,ADMINDIR)
GCMDATDEVDIR="/mnt/vale.2/GCMDAT_DEV"
GCMDATDIR="/mnt/vale.2/GCMDAT_V1.1"
ADMINDIR="$GCMDATDEVDIR/_ADMIN"

ALLFILES=‘find $GCMDATDIR -name *$VAR.*.$YEAR.lola.gz‘

TIMESTAMP=‘date‘

print "\n**********************************" >> $ADMINDIR/$INFOFILE
print "* " $TIMESTAMP >> $ADMINDIR/$INFOFILE
print "**********************************\n" >> $ADMINDIR/$INFOFILE

for FILE in $ALLFILES ; do
   print "\n" $FILE >> $ADMINDIR/$INFOFILE "\n"

   mknod theLolaInput p
   lola info theLolaInput >> $ADMINDIR/$INFOFILE &
   gzcat $FILE >> theLolaInput
   rm -f theLolaInput

done
```

A.1.2.2 checkFullYear

```
#!/usr/bin/ksh
# ----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#=============================================================================
#
#  Name    :  checkFullYear.ksh
#
#  Version :  1.0
#
#  Purpose :  check, if number of months modulo 12 is 0 (eg. only complete years)
#
#=============================================================================
#  Implementation and Revisions:  #
#-----------------------------------------------------------------------------
#
# Author Date Description
# --------------------
# rk 04/02/02 First implementation
# rk 14/03/02 counts records and years. division must be 12!
#
#=============================================================================
# read from command line

# set base path variables (GCMDATDIR,ADMINDIR)
GCMDATDIR="/mnt/vale.2/GCMDAT_DEV"
ADMINDIR="$GCMDATDIR/_ADMIN"

# set variables
ALLFILES=‘find $GCMDATDIR -name *.1013.lola.allyears.gz‘

function count_rec {
   mknod theLolaInput1 p
   lola nrec theLolaInput1 &
   gzcat $1 >> theLolaInput1
   rm -f theLolaInput1
}

function show_year {
   mknod theLolaInput2 p
   lola showyear theLolaInput2&
   gzcat $1 >> theLolaInput2
   rm -f theLolaInput2
}

function count_year {
      let first=0
      let last=0
      for YEAR in $ALLYEARS ; do
         if ((($first==0) || ($last==0))); then
            let first=$YEAR
    let last=$YEAR
         fi

         if (($first>$YEAR)); then
            let first=$YEAR
         fi

         if (($last<$YEAR)); then
            let last=$YEAR
         fi
      done
      let years=last-first+1
      print $years
}

   for FILE in $ALLFILES ; do
      let records=$(count_rec $FILE)
      ALLYEARS=$(show_year $FILE)
      let years=$(count_year)

      if (($records/$years!=12)) ; then
         print "$FILE has incomplete years! n=$n"
      else
         print "${FILE##/*/} has $years years in $records records!"
```

```
      fi
   done
```

A.1.2.3 Kelvin2Celsius

```
#!/usr/bin/ksh
# ----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#==============================================================================
#
#  Name    :  Kelvin2Celsius.ksh
#
#  Version :  1.0
#
#  Purpose :  changes Unit in datafile from degree Kelvin to Celsius
#
#==============================================================================
#  Implementation and Revisions:  #
#------------------------------------------------------------------------------
#
# Author   Date       Description
# ------   ----       -----------
# rk       07/12/01   First implementation
#
#==============================================================================
# read from command line
DIR=$1

# set base path variables (GCMDATDIR,ADMINDIR)
GCMDATDIR="/mnt/vale.2/GCMDAT_DEV"
ADMINDIR="$GCMDATDIR/_ADMIN"

# set variables
ALLLOLAFILES=`find $DIR -name *temp*lola`

   for FILE in $ALLLOLAFILES ; do
      mv $FILE $FILE.old
      lola -subc,273.15 $FILE.old $FILE
      rm $FILE.old
   done
```

A.1.2.4 Pa2mbar

```
#!/usr/bin/ksh
# ----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#==============================================================================
#
#  Name    :  Pa2mbar.ksh
#
#  Version :  1.0
#
#  Purpose :  changes Unit in datafile from degree Kelvin to Celsius
#
#==============================================================================
#  Implementation and Revisions:  #
#------------------------------------------------------------------------------
#
# Author   Date       Description
# ------   ----       -----------
# rk       07/12/01   First implementation
#
#==============================================================================
# read from command line
DIR=$1

# set base path variables (GCMDATDIR,ADMINDIR)
GCMDATDIR="/mnt/vale.2/GCMDAT"
ADMINDIR="$GCMDATDIR/_ADMIN"

# set variables
ALLLOLAFILES=`find $DIR -name *slp*lola`

   for FILE in $ALLLOLAFILES ; do
      mv $FILE $FILE.old
      lola -divc,100.0 $FILE.old $FILE
      rm $FILE.old
   done
```

A.1.3 Insertion of Data into the Database Development Environment

A.1.3.1 insGFDB

```
#!/usr/bin/ksh
# ----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#==============================================================================
#
#  Name    :  insGFDB.ksh
#
#  Version :  1.0
#
#  Purpose :  Insert Datasets into GFDB (Geophysical Fields Data Base)
#
```

```
#==============================================================================
#  Implementation and Revisions:  #
#------------------------------------------------------------------------------
#
# Author   Date       Description
# ------   ----       -----------
# rk       07/11/01   First implementation
#
#==============================================================================
# read from command line
MODEL=$1
LEVEL=$2

# set base path variables (GCMDATDIR,ADMINDIR)
GCMDATDIR="/mnt/vale.2/GCMDAT_DEV"
ADMINDIR="$GCMDATDIR/_ADMIN"
MODELDIR="$GCMDATDIR/$MODEL"

# set variables
SCENDIR=`find $MODELDIR -name $MODEL* -type d | grep $MODELDIR/`
TIMESTAMP=`date`

print "\n**********************************" >> $MODELDIR/insGFDB.log
print "* " $TIMESTAMP >> $MODELDIR/insGFDB.log
print "**********************************\n" >> $MODELDIR/insGFDB.log

# look for scenarios in model...
for DIR in $SCENDIR ; do
   print "SCENDIR:" $DIR >> $MODELDIR/insGFDB.log
   SPATIAL_RES=${DIR##/*/*.}
   print "SPATIAL_RES:" $SPATIAL_RES >> $MODELDIR/insGFDB.log
   ALLFILES=`find $DIR -name *$SPATIAL_RES.*.$LEVEL.lola.gz`
   print "ALLFILES:" $ALLFILES >> $MODELDIR/insGFDB.log
   # for all files in scenario...
   for FILE in $ALLFILES ; do
      print "\tFILE:" $FILE >> $MODELDIR/insGFDB.log

      # gunzip and rename datafile
      gunzip $FILE
      FILE=${FILE%.*}

      set -A YEARS $(split2annual.ksh $FILE)
      updateDataSets.ksh $FILE ${YEARS[0]} ${YEARS[1]}

      # gzip datafile
      gzip $FILE
   done
   updateDataLocations.ksh $DIR
   updateDataSources.ksh $MODEL $DIR
done
```

A.1.3.2 split2annual

```
#!/usr/bin/ksh
# ------------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#==============================================================================
#
#  Name    :  insGFDB.ksh
#
#  Version :  1.0
#
#  Purpose :  splits a multi-year-file into annual-files (ie. each new file
#             contains data from one year)
#
#==============================================================================
#  Implementation and Revisions:  #
#------------------------------------------------------------------------------
#
# Author Date Description
# ---------------------
# rk 07/11/01 First implementation
# rk 06/12/01 GCMDATDIR changed (/export/home -> mnt/vale.2)
#
#==============================================================================
# read from command line
DATAFILE=$1

# set base path variables (GCMDATDIR,ADMINDIR)
GCMDATDIR="/mnt/vale.2/GCMDAT_DEV"
ADMINDIR="$GCMDATDIR/_ADMIN"

# create a list of included years
YEARLIST=`lola showyear $DATAFILE`

# write one file for every year
let COUNT=0
for YEAR in $YEARLIST ; do
   TEMPLIST[$COUNT]=$YEAR
   YEARFILE=`print $DATAFILE | sed -e "s/lola/$YEAR.lola/g"`

   lola selyear,$YEAR,$YEAR $DATAFILE $YEARFILE
   gzip -f $YEARFILE

   let COUNT=$COUNT+1
done

# return (quasi) to insGFDB.ksh
print ${TEMPLIST[0]}
print ${TEMPLIST[$COUNT-1]}
```

A.1.3.3 updateDataSets

```
#!/usr/bin/ksh
# ----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#==============================================================================
#
#  Name    :  updateDatasets.ksh
#
#  Version :  1.0
#
#  Purpose :  update entries in GFDB.Datasets (each multiyear-datafile gets
#             one entry)
#
#==============================================================================
#  Implementation and Revisions:  #
#------------------------------------------------------------------------------
#
# Author Date Description
# --------------------
# rk 20/11/01 First implementation
# rk 06/12/01 GCMDATDIR changed (/export/home -> /mnt/vale.2)
#
#==============================================================================
# read from command line
DATAFILE=$1
FIRSTYEAR=$2
LASTYEAR=$3

# set base path variables (GCMDATDIR,ADMINDIR,GFDBDIR)
GCMDATDIR="/mnt/vale.2/GCMDAT_DEV"
ADMINDIR="$GCMDATDIR/_ADMIN"
GFDBDIR="$GCMDATDIR/_GFDB"

# split path (CCSR_CTRL.monthly.mean.r72x36.temp.1013.lola)
DUMMY=${DATAFILE%.*} # - gz
DUMMY=${DATAFILE%.*} # - lola
LEVEL=${DUMMY##/*/*.} # extract LEVEL
DUMMY=${DUMMY%.*} # - LEVEL
VARIABLE=${DUMMY##/*/*.} # extract VARIABLE
DUMMY=${DUMMY%.*} # - VARIABLE
GRID=${DUMMY##/*/*.} # extract GRID TYPE
DUMMY=${DUMMY%.*} # - GRID TYPE
STAT=${DUMMY##/*/*.} # extract STATISTIC
DUMMY=${DUMMY%.*} # - STATISTIC
TEMP_RES=${DUMMY##/*/*.} # extract TEMP_RES
DUMMY=${DUMMY%.*} # - TEMP_RES

#set -A LONGITUDES ‘lola griddes $DATAFILE | grep western‘
#LON=${LONGITUDES[0]}
LON_MIN="2.5"
LON_MAX="357.5"
#set -A LATITUDES ‘lola griddes $DATAFILE | grep northern‘
#LAT=${LATITUDES[0]}
LAT_MIN="-87.5"
LAT_MAX="87.5"

# update GFDB.Datasets
SCENDIR=‘dirname $DATAFILE‘
SCEN_ID=‘basename $SCENDIR‘
SCEN_ID=${SCEN_ID%.*}

print $SCEN_ID"\t" \
      $TEMP_RES"\t" \
      $STAT"\t" \
      $VARIABLE"\t" \
      $LEVEL"\t" \
      $FIRSTYEAR"\t" \
      $LASTYEAR"\t" \
      $GRID"\t" \
      $LON_MIN $LON_MAX"\t"  \
      $LAT_MIN $LAT_MAX"\t" \
      "\""$SCEN_ID".doc\"" \
      >> $GFDBDIR/GFDB.Datasets
```

A.1.3.4 updateDataLocations

```
#!/usr/bin/ksh
# ----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#==============================================================================
#
#  Name    :  updateDataLocations.ksh
#
#  Version :  1.0
#
#  Purpose :  update entries in GFDB.DataLocations (each scenario gets an entry)
#
#==============================================================================
#  Implementation and Revisions:  #
#------------------------------------------------------------------------------
#
# Author Date Description
# --------------------
# rk 20/11/01 First implementation
# rk 06/12/01 GCMDATDIR changed (/export/home -> /mnt/vale.2)
#
#==============================================================================
# read from command line
SCENDIR=$1
```

```
# set base path variables (GCMDATDIR,ADMINDIR,GFDBDIR)
GCMDATDIR="/mnt/vale.2/GCMDAT_DEV"
ADMINDIR="$GCMDATDIR/_ADMIN"
GFDBDIR="$GCMDATDIR/_GFDB"

# get grid type and scenario_id from SCENDIR (<model_id>_<scen_id>.<grid_type>)
SPATIAL_RES=${SCENDIR##/*/*.}
SCEN_ID=‘basename $SCENDIR .$SPATIAL_RES‘
SCENDIR=${SCENDIR%.*}
DATA_STRUCTURE="flat"

# update GFDB.DataLocations
print $SCEN_ID"\t\""$SCENDIR"\"\t"$DATA_STRUCTURE >> $GFDBDIR/GFDB.DataLocations
```

A.1.3.5 updateDataSources

```
#!/usr/bin/ksh
# -----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#=============================================================================
#
#  Name    :  updateDataSources.ksh
#
#  Version :  1.0
#
#  Purpose :  update entries in GFDB.DataSources (each scenario gets an entry)
#
#=============================================================================
#  Implementation and Revisions:  #
#-----------------------------------------------------------------------------
#
# Author Date Description
# --------------------
# rk 20/11/01 First implementation
# rk 06/12/01 GCMDATDIR changed (/export/home -> /mnt/vale.2)
#
#=============================================================================
# read from command line
MODEL=$1
SCENDIR=$2

# set base path variables (GCMDATDIR,MODELDIR,ADMINDIR,GFDBDIR)
GCMDATDIR="/mnt/vale.2/GCMDAT_DEV"
ADMINDIR="$GCMDATDIR/_ADMIN"
GFDBDIR="$GCMDATDIR/_GFDB"

# get grid type from SCENDIR
SPATIAL_RES=${SCENDIR##/*/*.}
SCEN_ID=‘basename $SCENDIR‘
SCEN_ID=${SCEN_ID%.*}

# read last line from GFDB.DataSources
set -A LASTLINE ‘egrep ’^[]*[0-9]+’ $GFDBDIR/GFDB.DataSources | tail -1‘
# and create number of new data source
let NUM=${LASTLINE[0]}+1

DESCR="http://www.giub.unibe.ch/~kernen/gcm/overview_bdy.html#"$MODEL

# update GFDB.DataSources
print $NUM"\t"$SCEN_ID"\t\""$DESCR"\"" >> $GFDBDIR/GFDB.DataSources
```

A.1.4 Visual Testing

A.1.4.1 Make_TestAtlas

```
#!/usr/bin/ksh
# -----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#=============================================================================
#
#  Name    :  Make_TestAtlas.ksh
#
#  Version :  1.0
#
#  Purpose :  Make a TestAtlas for visual control of the GCMDAT
#
#=============================================================================
#  Implementation and Revisions:  #
#-----------------------------------------------------------------------------
#
# Author Date Description
# --------------------
# rk 30/08/02 first implementation
#
#=============================================================================
paramfile=$2

# call to program from GFDBTOOLS package
MakeGFDBAtlas $paramfile
```

A.1.4.2 MGA.CTRL.TestGCMDAT

```
#=============================================================================
#
#  Name   :  Sample control file for program MakeGFDBAtlas
#
```

```
#  Purpose:  Specifies parameters for the production of a web-viewable
#            digital atlas of one or several data sets from a
#            Geophysical Fields Data Base (GFDB).
#
#==============================================================================

# GFDBDIR=/mnt/vale.2/GCMDAT_V1.0/_GFDB

ATLAS_DESCRIPTOR        "GCM Atlas V1.0 Tests"
ATLAS_TARGET_DIR        "/home/klimet/kernen/pwww/gcm/TestAtlas"

#DATA_SET_NUMS           1
DATA_SET_NUMS           13 14 15


SEASONS                 Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

  MAPS_ANALYSIS         Globe     longTermMean 1971 1971 producePlots

#==============================================================================
```

A.1.5 Releasing of GCMDAT

A.1.5.1 Make_GCMDAT_Release

```
#!/usr/bin/ksh
# ----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#==============================================================================
#
#  Name    :  Make_GCMDAT_Release.ksh
#
#  Version :  1.0
#
#  Purpose :  Extrakt Release from Developer Version of GCMDAT
#
#==============================================================================
#  Implementation and Revisions:  #
#------------------------------------------------------------------------------
#
# Author Date Description
# --------------------
# rk 18/12/01 first implementation
# rk 03/07/02 datafiles are now being moved (they were copied before)
# querystring in find commmand (before moving the datafiles)
# is now *.lola.gz (was *.1013.????.lola.gz before)
#
#=============================================================================
# get parameterfile from commandline
VERSION=$1

# set path variables
GCMDATDEVDIR=/mnt/vale.2/GCMDAT_DEV
GCMDATDIR=/mnt/vale.2/GCMDAT_$VERSION

   if [ ! -d $GCMDATDIR ]
      then
      print $GCMDATDIR " does not exist! Create manually!"
   fi

# look for models in DEVLEOPER Version
unalias ls
MODELDIR=`ls $GCMDATDEVDIR | grep -v ^_`

# selection of models
# let user select models to be included
#STRING=selected
let SEL=1
let NSEL=1
for DIR in $MODELDIR ; do
   read REPLY?"include $DIR ? (y/n) "
   case $REPLY in
      y )
        print "$DIR is included in $GCMDATDIR"
SELECTED[$SEL]=$DIR
let SEL=$SEL+1
print 'selected' >> $GCMDATDEVDIR/_GFDB/not_selected.txt;;
      n )
        print "$DIR is NOT included in $GCMDATDIR"
print $DIR >> $GCMDATDEVDIR/_GFDB/not_selected.txt;;
      * )
        print "$DIR is NOT included in $GCMDATDIR"
print $DIR >> $GCMDATDEVDIR/_GFDB/not_selected.txt;;
   esac
done

# copy datafiles
# for every selected model
let COUNT=1
while [ $COUNT -lt $SEL ]; do
   MODEL=${SELECTED[COUNT]}

   if [ ! -d "$GCMDATDIR/$MODEL" ]
      then
      mkdir $GCMDATDIR/$MODEL
   fi

   # for all scenario directories in selected model
   SCENDIR=`ls "$GCMDATDEVDIR/$MODEL" | grep -v ^_ | grep $MODEL`
   for SCEN in $SCENDIR ; do
      if [ ! -d $GCMDATDIR/$MODEL/$SCEN ]
```

```
         then
         mkdir $GCMDATDIR/$MODEL/$SCEN
      fi

      # move all *lola.gz from GCMDAT_DEV to RELEASE-DIRECTORY
      ALLFILES=`find "$GCMDATDEVDIR/$MODEL/$SCEN" -name *.lola.gz`
      print "copy datafiles for $SCEN"
      for FILE in $ALLFILES ; do
         FILE=${FILE##/*/}
         mv $GCMDATDEVDIR/$MODEL/$SCEN/$FILE $GCMDATDIR/$MODEL/$SCEN/$FILE
      done
   done
   let COUNT=$COUNT+1
done

# copy gfdb-files and documentation
# only entries of included models, path changed from GCMDAT_DEV to GCMDAT_V1.0

   if [ ! -d $GCMDATDIR/_GFDB ]
      then
      mkdir $GCMDATDIR/_GFDB
   fi

/usr/xpg4/bin/grep -v -f $GCMDATDEVDIR/_GFDB/not_selected.txt \
$GCMDATDEVDIR/_GFDB/GFDB.DataLocations | sed -e "s/GCMDAT_DEV/GCMDAT_$VERSION/" > \
$GCMDATDIR/_GFDB/GFDB.DataLocations

/usr/xpg4/bin/grep -v -f $GCMDATDEVDIR/_GFDB/not_selected.txt \
$GCMDATDEVDIR/_GFDB/GFDB.DataSources | sed -e "s/GCMDAT_DEV/GCMDAT_$VERSION/" > \
$GCMDATDIR/_GFDB/GFDB.DataSources

/usr/xpg4/bin/grep -v -f $GCMDATDEVDIR/_GFDB/not_selected.txt \
$GCMDATDEVDIR/_GFDB/GFDB.Datasets | sed -e "s/GCMDAT_DEV/GCMDAT_$VERSION/" > \
$GCMDATDIR/_GFDB/GFDB.Datasets

rm $GCMDATDEVDIR/_GFDB/not_selected.txt

cp -r $GCMDATDEVDIR/_DOCU $GCMDATDIR
```

A.1.6 Final Tests

A.1.6.1 TestDB

```
#!/usr/bin/ksh
# -----------------------------------------
# Climatology and Meteorolgy Research Group
# Department of Geography
#==============================================================================
#
#  Name    :  TestDB.ksh
#
#  Version :  1.0
#
#  Purpose :  Test GCMDAT
#
#==============================================================================
#  Implementation and Revisions:  #
#------------------------------------------------------------------------------
#
# Author Date Description
# ---------------------
# rk 21/12/01 first implementation
#
#==============================================================================
GCMDATDIR=$1

# set base path variables (GCMDATDIR,ADMINDIR)
GFDBDIR="$GCMDATDIR/_GFDB"
DOCUDIR="$GCMDATDIR/_DOCU"

grep '"' $GFDBDIR/GFDB.Datasets > $DOCUDIR/Datasets.txt

# for every line in $DOCUDIR/Datasets.txt extract data from GFDB
while read LINE; do
   GFDB -b . `print $LINE | awk '{ print $1,$2,$3,$8,$4,$5,$6,$7,$9,$10,$11,$12 }'`
done < $DOCUDIR/Datasets.txt
```