# XMM

# User Guide for the Mirror Metrology Database

XMM-PS-TN-10 Issue 1.0

Written by: D. Holling

April 12, 1996

Revision history

| Revision number | Date | Revision author | Comments |
|---|---|---|---|
| Issue 1.0 | April 12, 1996 | D. Holling | Initial version |

# Contents

# 1 Introduction

This document is a User's Guide to the software developed to allow access and processing of mirror metrology data delivered from the manufacturer. This software will be referred to throughout this document as the metrology application tool. In addition, information relating to the structure and content of the mirror metrology database is discussed.

The information in this document relates to version 1.2 of the software. Although strictly not part of a User Guide, information about the software structure, installation, configuration and internal data files is included in this document for completeness.

It is assumed that the reader is familiar with the basics of the Unix operating system, that the reader is working under a flavour of the Unix operating system and that IDL is correctly installed and operational on the reader's machine.

Commands to be typed at the Unix command line will be shown `using this typewriter text style`. Filenames and file pathnames, and text seen on the computer display but not entered by the user will be shown using this Sans Serif text style.

# 2 Software Installation And Environment

## 2.1 Metrology Application Tool Software Files and Location

The metrology database data processing tool is written entirely in IDL, a functionally rich data analysis and display language. All files for a given, official version of the metrology application tool can be found in the root directory :

/usr4/users/dholling/xmm/mirrordb/release

Each version has a sub-directory of this root named :

vn-m

where n and m are integers indicating the main and sub-release numbers respectively (e.g. version 1.2 is found in the sub-directory labelled v1-2).

Each version sub-directory contains the directory tree and files necessary to build the version of the code, plus a tar file, labelled xmmdb.n-m, where n and m are the version numbers appropriate to the release, which combines the full release directory tree and files using the standard Unix tar command.

The source files comprising version 1.2 of the metrology application tool are placed in the directory tree shown below :

```
/usr4/users/dholling/xmm/mirrordb/release/v1-2
        main.idl
        metrology.mak
      -> /procedure
              action.idl
              combine.idl
              display.idl
              graph.idl
              parseact.idl
              preprocess.idl
              pwrdensity.idl
              read.idl
              refractive.idl
              surface.idl
              usr_routines.idl
      -> /status
              action.int
              action.stat
              combine.stat
              display.int
              display.stat
```

```
            global.int
            global.stat
            graph.int
            graph.stat
            preprocess.int
            preprocess.stat
            pwrdensity.stat
            read.int
            read.stat
            refractive.stat
            usr_routines.stat
    -> /structure
            act-list.stct
            action.stct
            config.stct
            curve.stct
            data-list.stct
            data-record.stct
            file-header.stct
            graph.stct
    -> /data
            inst-action.smp
            inst-data-config.smp
    -> /template
            action.smp
            data-config.smp
    -> /common
            colour.com
    -> /henke_code
            henke2.xdr
```

The following text contains a brief description of the files, directories and their contents.

main.idl : This *file* contains the top level IDL routine used to run the metrology application tool.

metrology.mak : This *file* contains the IDL batch makefile instructions used to compile and run the metrology application tool within IDL.

procedure : This *directory* contains the IDL source code of the routines used within the metrology application tool. The routines are divided across a number of separate module files, each containing the definitions of functionally similar routines. For example, all routines used to read metrology database data files are held in the module read.idl. Each routine within a module file is named with a three letter prefix indicating the module file name containing the function source code (this aids finding the source module of a routine when it is being called from a different module).

status : This *directory* contains the IDL include file definitions of the constant variables and module interface values used within the module files of metrology application tool. The interface constant values are stored in the files suffixed *.int whilst the module constants are stored in the files suffixed *.stat.

template : This *directory* contains two files which are self-documenting templates for the action and data definition data files used by the metrology application tool. They contain commented definitions of all valid fields accepted by the metrology application tool either in source data or action (data processing) definition.

structure : This *directory* contains the structure definitions used within the metrology application tool routine code.

data : This *directory* contains the action and data definition data files used by the metrology application tool on each run.

common : This *directory* contains the common block definitions used in the metrology application tool.

henke_code : This *directory* contains the Henke X-ray scattering coefficients used in the calculation of complex refractive indices and reflectivities.

This kind of file organisation and directory structure is found in each release of the metrology application tool.

## 2.2 Metrology Application Tool Software Installation

The metrology application tool is installed in a target root directory by unpacking the tar file containing the given release (described in the sub-section above) in any directory chosen to be the root of the installation.

1. Copy the version tar file from the official release directory into the directory **above** the target root directory (/data/transport/dholling/software/mmdb, for example). Type :

```
unix-prompt> cp /usr4/users/dholling/xmm/mirrordb/release/v1-2/xmmdb.1-2
/data/transport/dholling/software
```

2. Change directory to the installation root directory by typing :

```
unix-prompt> cd /data/transport/dholling/software/mmdb
```

3. Unpack the tar file (creating automatically the directory tree) by typing :

```
unix-prompt> tar xvf ../xmmdb.1-2
```

4. Edit the metrology database root pathname in the file read.stat to be that appropriate

to the local installation (db-data-root for example) :

```
unix-prompt> edt status/read.stat
```

Type in the following and save the changed file :

```
strMetrologyRoot = '/db-data-root'
```

5. Edit the Henke scattering coefficient data pathname in the file refractive.stat to be that appropriate to the local installation (db-henke-root for example) :

```
unix-prompt> edt status/refractive.stat
```

Type in the following and save the changed file :

```
strOscFilename = '/db-henke-root'
```

## 2.3 The Metrology Application Tool System Resources

There are no specific resources required by the metrology application tool apart from the file system space necessary to store the application source files, and the resources required to run IDL. The reader is referred to the local IDL system administrator for resource problems.

## 2.4 Running The Metrology Application Tool

1. Change directory to the root directory of the metrology database tool.

```
unix-prompt> cd /data/transport/dholling/software/mmdb
```

2. Start IDL by typing (it is assumed that IDL is correctly installed and available on the reader's machine) :

```
unix-prompt> idl
```

3. In the main IDL command window appearing, compile and run the metrology database tool by typing :

```
idl-prompt> @metrology.mak
```

metrology.mak is a batch command file that invokes the main top-level IDL routine of the metrology database application, stored in the file /data/transport/dholling/software/mmdb/main.idl, after first compiling all the code in the IDL environment.

4. Subsequent re-runs of the metrology application code may be made by running step 3

again at the IDL prompt.

# 3 Metrology Application Tool Functionality

## 3.1 Top Level Structure And Description

All the metrology application tool functionality has been integrated into one top level IDL routine that is completely driven by the input of two data files. Once the two data files have been correctly set-up, there is no need of interactive user input during the running of the tool.

The two configurable data files are used to control which data sets from the metrology database are initially used and the processing performed on those data sets.

The data files are named inst-data-config.smp and inst-action.smp for the data set specification and the data processing specification, respectively. These files are stored in the metrology application tool sub-directory data, as described in Section 2.

The data processing instructions are split into one or more atomic steps called **actions**. Each action is described in the following sub-section. Each action is defined in the action configuration file using a number of fields to be discussed in a later sub-section.

Each data set specified in the data configuration file is assumed to be present in the metrology database. Each of these data sets is assigned a unique integer identifier, monotonically increasing from 1, on reading in the data. Each action specifies one or more target (source) data set identifiers which it uses to execute the action. The result of the action processing is given a user specified identifier, which should be unique.

The actions are executed in the order of specification in the action configuration file, first to last. Thus, an action may create a new data set which is used as the input to subsequent actions.

It is the user's responsibility to ensure that the data set identifiers of each data set is unique.

The actions are totally generic, so that the user, by specifying the order of the actions and the data on which those actions are performed, totally dictates what happens (rather than be bound by a limited number of routines that just read and display specific sets of data).

There is no limit of the number of actions that can be performed on the data, or the order in which the actions are performed i.e you can average three data series and fractal fit them, or combine three series and fractal fit them ...etc. The user controls, given the available data, what is processed, stored and displayed.

The main routine run in the IDL environment returns a value of 0 if the data files and data processing actions have been successfully read in and completed. Any errors during the processing result in an immediate abort of the run with a negative integer error code being returned. Unfortunately there has not been time to implement a sensible error and process logging mechanism that would allow easy analysis of the fault, but this may be easily done with two to three days effort in the future, if desired. Currently one has to resort to tracking the error codes documented in the IDL status files and placing debug **PRINT** statements in the code.

## 3.2 Application Tool Return Error Codes

The metrology application tool top level routine returns a code 0 if it completes with no problems or errors, else a negative integer is returned indicating where the error or problem occured in the code.

The error codes and the possible reasons for a problem are discussed below.

| Error Code | Reasons For Problem |
|---|---|
| -1 | There has been an error reading the data configuration file. Check the format and record content of the data configuration file. Check that the data configuration file exists and is in the correct directory location. |
| -2 | There has been an error pre-processing some of the fields read in from the data configuration file. There are probably errors contained in the data records in the data configuration file (parameters having incorrect values etc.). |
| -3 | There has been an error reading the action configuration file. Check the format and record content of the action configuration file. Check that the action configuration file exists and is in the correct directory location. |
| -4 | There has been error processing the actions read in from the directory file. Check that the data types being manipulated are in the correct internal array format for the action. Check that the chain of SOURCE ID identifiers both exist at the time of the action invocation and are unique. |
| -5 | There have been problems reading in the source data specified in the records contained in the data configuration file. Check that all specified data exists in the metrology database (as data files are opened for reading, the full pathname is printed on the IDL command window). Check that the files have the correct format. |

If these checks do not show an obvious problem, further checks within the debug environment of IDL will be required using an experienced IDL developer.

## 3.3 Description Of Internal Storage Of Metrology Data

This subsection describes the pre-processing applied to the metrology data read in by the metrology application tool (if any), and the internal storage structure in the software in which the data is kept. Understanding this mechanism aids understanding which actions it is sensible to apply to which data sets.

Note that this section refers to the internal storage mechanism of the data and does not describe each data type. The reader is referred to Section 5 for details of the units of the data types.

### 3.3.1 BSDF Data

There is no pre-processing done to the data read in from the metrology database.

This data is stored in a 3 dimensional array sized as follows :

arrOut(num elevations, num azimuths, 3)

where,

arrOut(i,j,2) contains the BSDF values across the elevation/azimuth grid

arrOut(i,j,0) contains the elevation angle values across the elevation/azimuth grid

arrOut(i,j,1) contains the azimuth values angle across the elevation/azimuth grid

The number of elevation and azimuth angle points on the grid varies according to the data read in as described the later section.

### 3.3.2 Screening Device Focal Point Data

There is no pre-processing done to the data read in from the metrology database.

This data is stored in a 2 dimensional array structured as follows :

arrOut(*, 0) = z coordinate values
arrOut(*, 1) = $X_{av}$ for each z
arrOut(*, 2) = $Y_{av}$ for each z
arrOut(*, 3) = HEW for each z
arrOut(*, 4) = 90EW for each z

where,

z = focal plane coordinate [mm]

$X_{av}$ = X coordinate of the average focus [mm]
$Y_{av}$ = Y coordinate of the average focus [mm]
HEW = Half Energy Width [arcsec]
90EW = Width of the cone including 90% of the energy [arcsec]

### 3.3.3 Screening Device Error Series Coefficient Data

There is no pre-processing done to the data read in from the metrology database.

This data is stored in a 2 dimensional array structured as follows :

arrOut(n, 0) = En,0,c
arrOut(n, 1) = En,1,c
arrOut(n, 2) = En,2,c
arrOut(n, 3) = En,3,c
arrOut(n, 4) = En,0,s
arrOut(n, 5) = En,1,s
arrOut(n, 6) = En,2,s

where, En,m,c is the $n^{th}$ order Legendre, $m^{th}$ order Fourier error coefficient of cosine term and En,m,s is the $n^{th}$ order Legendre, $m^{th}$ order Fourier error coefficient of sine term in the orthonormal surface error expansion.

### 3.3.4 PROMAP Shell And Mandrel Data

After the raw data is read in from the metrology database, a parabola is best fitted to the data and subtracted from the raw data.

The data is stored in a 2 dimensional array arrOut. The normal dimensions are arrOut(*, 11) This corresponds to 10 data series plus one set of independent variable values, each consisting of a number of points which vary according to the resolution of the PROMAP measurement.

The $i^{th}$ independent variable value of ALL rows is placed in the array element arrOut(i - 1, 0). Note that the independent variable value is relative to the start of the PROMAP measurement according to the spatial resolution defined in the PROMAP file header.

The $i^{th}$ PSDF value of the kth data series row is placed in the array element arrOut(i - 1, k).

It is assumed that the same spatial displacement steps apply to all rows.

#### 3.3.5 PROMAP PSDF Data

The independent variable value in the PSDF measurement is reciprocated to obtain the frequency space equivalent after the data is read in from the metrology database.

The data is stored in a 2 dimensional array `arrOut`. The normal dimensions are `arrOut(640, 11)` This corresponds to 10 data series (and one set of independent variable values), each consisting of 640 points.

The i$^{th}$ frequency value (independent variable) of ALL rows is placed in the array element `arrOut(i - 1, 0)`.

The i$^{th}$ PSDF value of the kth data series row is placed in the array element `arrOut(i - 1, k)`.

It is assumed that the same frequency steps apply to all rows.

#### 3.3.6 UPMC Axial Profile Data

The independent variable value, i.e. axial coordinate, is shifted in value so that the origin lies in intersection of parabolic and hyperbolic sections of the mirror after the data is read in from the metrology database.

The data is stored in a 2 dimensional array, sized according to the number of lines in the file being read. The values arrOut(*,0) contain the x position of the UPMC measurement and the values arrOut(*, 1) contain the UPMC height measurement.

#### 3.3.7 UPMC Azimuthal Profile Data

There is no pre-processing done to the data read in from the metrology database.

The data is stored in a 2 dimensional array, sized according to the number of lines in the file being read. The values arrOut(*,0) contain the azimuthal angle of the UPMC measurement and the values arrOut(*, 1) contain the UPMC height measurement.

#### 3.3.8 CMD Mandrel Data

There is no pre-processing done to the data read in from the metrology database.

The data is stored in a 1 dimensional array, sized according to the number of points stored in the CMD data file. **This will probably need changing if and when the use of CMD data is actually required in the metrology application tool.**

## 3.4 Description Of Functionality

### 3.4.1 Combine Two Data Series

It is possible to combine user-specified regions of two data sets into one. The desired regions of each data set are selected by means of a condition string applied to the independent variable value of each data set.

Each data set should be in the form of a two-dimensional array, A(i,j), where the independent variable values are stored in the position A(*,0). A(*, j), where j > 0 store the dependent variable values of one or more data series. Only the first dependent variable data series in each input data set array is combined i.e. all points in A(*, j), where j > 1 are ignored by this operation.

The output is a two dimensional array B(l,m), where B(*,0) stored the ordered independent variable values over the combined region, and B(*,1) store the corresponding dependent variable values.

### 3.4.2 Average Data Series

It is possible to average one or more data sets. It is assumed that each data set averaged has an identical range and number of independent variable values (i.e. each data set has identical independent variable values).

Each data set should be in the form of a two-dimensional array, A(i,j), where the independent variable values are stored in the position A(*,0). A(*, j), where j > 0 store the dependent variable values of one or more data series. All dependent variable data series in each input data set array are averaged.

The output is a two dimensional array B(l,m), where B(*,0) stores the independent variable values, and B(*,1) store the corresponding averaged dependent variable values.

### 3.4.3 Fractal Fit Data Series

It is possible to calculate a best fit fractal function for a data set. The following function is used to fit the two variables, $K$ and $n$, to the data set :

$$Kx^n$$

where $x$ is the independent variable value.

The input data set should be in the form of a two-dimensional array, A(i,j), where the independent variable values are stored in the position A(*,0). A(*, j), where j > 0 store the dependent variable values of one or more data series. Only the first dependent data series is fitted i.e. the data in A(*, 1).

The output is a two component, one dimensional array, $[K, n]$.

### 3.4.4 Generate Mirror Surface Error Corrections

It is possible to generate error expansion terms about an ideal mirror surface of user specified radius for a given order of a Fourier and Legendre expansion term. This MUST use data from a screening device orthogonal error series data file as source data as specific positions of error coefficients are assumed in the data stream.

The z axis is taken as the axis of the cylindrical shell. The surface error correction is calculated for a specific order of term in the Fourier and Legendre components. This is done for 21 equally spaced azimuthal angle values, at each of 21 equally spaced relative axial coordinate values along the axial length of the mirror from -1 to 1 (-1 corresponds to the entrance to the parabolic section and 1 corresponds to the exit of the hyperbolic section).

The output is a three dimensional array of size B(2l,21,3), where B(*,*,0) stores the x values across a regular grid, B(*,*,1) stores the y values across a regular grid, and B(*,*,2) stores the z axial coordinate values.

B(0:20,0,2) contains the first fixed z value i.e. -1, B(0:20,20,2) contains the final z value, 1.

B(0:20,0,0) contains the x coordinates in the azimuthal plane at the first fixed z = -1 corresponding to each of the azimuthal angles. B(0:20,0,1) contains the corresponding y coordinate in the azimuthal plane.

### 3.4.5 Grid Data Series

This combines multiple data sets to generate a 'grid' of data sets inside a single multiple dimensional array. It is assumed that each set gridded has a common independent variable value and number of points in the set. The specified data sets are taken in order and concatenated into a single array.

Each input data set should be in the form of a two-dimensional array, A(i,j), where the independent variable values are stored in the position A(*,0). A(*, j), where j $>$ 0 store the dependent variable values of one or more data series. If a contributing data set has more than one series in it then all series are gridded in order of storage in the data set.

The output is a two dimensional array B(l,m), where B(*,0) stored the independent variable values, and B(*,j), where j $>$ 0, store the corresponding gridded dependent variable values. j is dimensioned to be the cumulative sum of all the data series in the input data arrays.

This functionality may be used, for example, to combine 35 separate UPMC axial data sets for a given mirror shell (each measurement set is stored separately in the metrology database) into a single array that can be stored as a 'grid' of measurements over the entire shell in a single file.

### 3.4.6 Generate A Data Series

This can be used to generate a series of points satisfying a given function. The data series generated has equally spaced points over a user specified range of values of independent variable (x-axis). Currently points satisfying the formula $Kx^n$ are supported.

The parameters $K$ and $n$ are supplied in the input/source data series, which is assumed to be a 1 dimensional array of the form $[K, n]$.

The output is a two dimensional array B(l,2), where B(*,0) stored the independent variable values, and B(*,1) store the corresponding generated data series values.

### 3.4.7 Save Data Series

It is possible to save data sets held in the internal array structure into a user specified file. An internal database format is used which can only be read using the RETRIEVE action. Any array up to and including dimension three can be saved.

For an array of the form A($n_1$, $n_2$, $n_3$), the convention adopted is that the leftmost index is the most rapidly changing, and the rightmost index is the least rapidly changing.

The format of the saved data file for a 3-dimensional array, the i$^{th}$ dimension containing $n_i$ points, is :

```
 Total number of dimensions (3)
Number of points in first dimension (n1)
⋮
Number of points in 3rd dimension (n3)
Point A(0, 0, 0)
Point A(1, 0, 0)
⋮
Point A(n1 - 1, 0, 0)
Point A(0, 1, 0)
⋮
Point A(n1 - 1, n2 - 1, 0)
Point A(0, 0, 1)
⋮
Point A(n1 - 1, n2 - 1, 1)
⋮
Point A(n1 - 1, n2 - 1, n3 -1)
```

The fastest changing index is first saved, followed by the next fastest, down to the slowest changing index.

### 3.4.8 Retrieve Saved Data Series

It is possible to retrieve saved data sets from a specified file. Do not use this function to read any metrology database files; only files created in the internal database format using SAVE should be read using this function.

An array identical to the one used to create the saved file is recreated in the metrology application tool.

### 3.4.9 Graphically Display Data Series

It is possible to display up to 10 data sets on a single two-dimensional graph. A choice of log-log, linear-linear, linear-log and log-linear axis sets is available to the user. The graph title, x and y axis labels and individual line labels are all user configurable. Each data set is plotted using a different colour and symbol style, and a key section is automatically provided using the user's line labels.

### 3.4.10 Display Data Series In 3-D

It is possible to display a data set on a single three-dimensional graph. The graph title, x, y and z axis labels are user configurable.

It is assumed that the data set to be plotted is in the form of a three dimensional array, where `B(*,*,0)` stores the x values across a regularly or near-regularly spaced grid, `B(*,*,1)` stores the y values across a regularly or near-regularly spaced grid, and `B(*,*,2)` stores the z (or function/surface) values at the specific points on the x/y grid.

## 3.5 Structure And Content Of The Action File

### 3.5.1 Structure Of The Action File

An example action file is shown in Appendix A. This contains, in the commented text, examples of the necessary fields and their values for each type of action supported by the metrology application tool.

The fields supported by the action file and their allowed values are discussed below in the following sub-section.

Each action consists of a set of information specified using a number of field records placed on one or more lines in the action configuration file. This information may be interleaved with any number of blank or comment lines. A comment line is any line that has the '#' character as the first non-whitespace character. The information relating to different actions is separated by a group separator character, '-'.

The action fields and their respective values in the action file are specified using entries having the general form :

```
field-name = field-value
```

The field name text must be in upper case and be the first non-whitespace text on a line. In addition, the field name text must agree in number of letters and spacing of words to the form listed below, and shown in the example action file in Appendix A.

Any field name not agreeing with the specified format is ignored. Multiple instances of the same field name record may appear in a specific action. However, in this instance, only the value in the last field name record is taken.

Two complete action records, including comments, are shown below to illustrate the use of group separator characters, comment characters and field records :

```
#
# EXAMPLE ACTION TO SAVE DATA READ IN
#

ACTION = SAVE
SOURCE IDS = 1
FILE PATHNAME = /usr4/users/dholling/xmm/mirrordb/dev/d4.tst
```

```
-

#
# EXAMPLE ACTION TO RETRIEVE DATA SAVED
#

ACTION = RETRIEVE

RESULT ID = 2
FILE PATHNAME = /usr4/users/dholling/xmm/mirrordb/dev/d4.tst

-
```

### 3.5.2 Field Records Supported In The Action File

This subsection lists and describes each action field name supported by the action file, its usage, and the allowed values of the field name.

Comment Character
Field Name : #
Field Value : None
Description : This is a comment character. All other text on the line after this character is ignored.

Group Separator Character
Field Name : -
Field Value : None
Description : This is an action record separator used to separate the field records of different actions.

Field Value Separator Character
Field Name : %
Field Value : None
Description : This is an field value record separator used to separate multiple values within a single field value record.

For example, three values in the field record SOURCE IDS are specified using the following mechanism :

SOURCE IDS = 1 % 2 % 3

Action Type Field Record
Field Name : ACTION =
Field Value : The field values supported are :

DISPLAY - display a data set either on a graphical or a 3-D surface.

FRACTAL FIT - fit a best fit fractal function to a data set. Generates/returns a 1-D array of two parameters [K,n] where K and n are fitted to the data set satisfying :

$$Kx^n$$

where $x$ is the independent variable value.

COMBINE - combine regions of two data sets into one.

SAVE - save data sets into a specified file. The internal database format is used which can only be read using the RETRIEVE action.

RETRIEVE - retrieve saved data sets from a specified file. Do not use this function to read any metrology database files.

EXT AVERAGE - average one or more data sets

GENERATE SURFACE - generate error expansion term about an ideal mirror surface for a given order of Fourier and Legendre expansion term. This MUST use orthogonal error series data file as source data.

GRID SERIES - generate a 'grid' of data sets. It is assumed that each set gridded has a common independent variable value and number of points in the set. The sets specified in the SOURCE IDS field are taken in order and concatenated into a single array. If a contributing data set has more than one series in it then all series are gridded in order of storage in the data set.

GENERATE DATA SERIES - this action can be used to generate a series of points satisfying a given function. The data series provides equally spaced points over a user specified range of values of independent variable (x-axis).

Description : This is the field record used to specify the type of action to execute.

Condition String Field Record
Field Name : CONDITIONS =

Field Value : The form of the field value record is :

cond-1 % cond-2

where cond-i is the condition string applicable to the i$^{th}$ source data set.

Each condition string has one of the following formats :

>= value
<= value
<= value1, >= value2

where value1 >= value2

Description : This specifies the two condition strings applied to the independent variable value in the data sets used in the COMBINE action, each condition being read as a string and separated by the '%' character. Note that if a data set contains multiple dependent variable series, only the first series is combined. Note also that the first data set specified in the SOURCE IDS list takes the first condition string in the CONDITIONS list. A condition string may be blank.

Data Series Field Record
Field Name : DATA SERIES TYPE =
Field Value : POWER - to fit the power series decribed below.

Description : This specifies the type of data series generated by the action GENERATE DATA SERIES. Values currently supported are :

> POWER - fits equally spaced points on a LOG10 scale using the formula $Kx^n$, where K and n are supplied by the SOURCE IDS and x is the independent variable value. Note that K is the first array element and n is the second array element in the data pointed to by SOURCE IDS.

Error Series Order Field Record
Field Name : EXPANSION =

Field Value : The field value of the error series order field has the following form :

symmetry term % legendre order % fourier order % radius

where :
Legendre order = order of Legendre polynomial of value 0 <= n <= 2
Fourier order = order of Fourier coefficient of value 0 <= m <= 3
symmetry term = c or s indicating cosine or sine part of the Fourier expansion respectively.
radius = radius (in mm) of the cylinder against which the surface error calculations are plotted.

Description : This specifies the error term required in the surface error calculation performed by the GENERATE SURFACE action. It has four fields as described above.

File Name Field Record
Field Name : FILE PATHNAME =
Field Value : Any valid file pathname may be entered. Note that quotes are not necessary in the string.

Description : This field is interpreted as a text string used to specify the full pathname of the internal database format file created/used by the RETRIEVE or SAVE actions in the internal database format.

Graph Type Field Record
Field Name : GRAPH TYPE =
Field Value : the allowed values of this field value record are :

LINEAR - plot a linear-linear (x/y) axis graph
LINEAR-LOG - plot a linear-log (x/y) axis graph
LINEAR-LOG - plot a log-linear (x/y) axis graph
LOG-LOG - plot a log-log (x/y) axis graph

Description : This specifies the type of graph on which to display the data for a graphical display. The allowed values are discussed above.

Label Field Record
Field Name : LABEL =
Field Value : Any valid alphanumeric string may be entered. Note that quotes are not necessary in the string.

Description : This field is interpreted as a text string. It is used to provide an array of labels for curves to be plotted on a single graphical display. **Consequently there may be more than one label field present in an action record definition.**

Note that it is the users responsibility to define an equivalent number of labels for the individual curves to be plotted (remember that some data sets contain multiple curves which each require an individual label). In addition the labels are applied in strict order i.e. first label to first curve etc.

Output Display Type Field Record
Field Name : OUTPUT TYPE =
Field Value : The allowed field record values are :

GRAPH - plot data on a graphical display
SURFACE - plot data as a 3-D surface

Description : This specifies the output display style for a display action. The allowed values are discussed above.

Data Series Range Field Record
Field Name : POINT RANGE =
Field Value : The field value record has the following form :

lower value % upper value

Description : This is a two record field specifying the lower and upper limit of independent variable used for the action GENERATE DATA SERIES.

Action Result Identifier Field Record
Field Name : RESULT ID =
Field Value : Integer value not previously used as a data set identifier.

Description : Integer value giving the metrology application tool identifier of data resulting from an action. It is the user's responsibility in configuring the action and data files to ensure that each different data series has a unique identifier.

Source Data Identifier Field Record
Field Name : SOURCE IDS =
Field Value : This field value record has the form :

id-1 % id-2 % . . . % id-n

where id-i is the integer data identifier of the $i^{th}$ data set to be used by the action.

Description : This specifies the identifiers of the data in the data list used as source data for an action. Each data set specified in the data configuration file is given a monotonically increasing integer identifier starting from 1. Subsequent data sets derived as the output to actions are also given user specified integer identifiers.

Title Field Record
Field Name : TITLE =
Field Value : Any valid alphanumeric string may be entered. Note that quotes are not necessary in the string.

Description : This field is interpreted as a text string used as the title of a graphical and surface display.

X-Axis Title Field Record
Field Name : XTITLE =
Field Value : Any valid alphanumeric string may be entered. Note that quotes are not necessary in the string.

Description : This field is interpreted as a text string used as the x-axis label of a graphical and surface display.

Y-Axis Title Field Record
Field Name : YTITLE =
Field Value : Any valid alphanumeric string may be entered. Note that quotes are not necessary in the string.

Description : This field is interpreted as a text string used as the y-axis label of a graphical and surface display.

Z-Axis Title Field Record
Field Name : ZTITLE =
Field Value : Any valid alphanumeric string may be entered. Note that quotes are not necessary in the string.

Description : This field is interpreted as a text string used as the z-axis label of a surface display.

## 3.6 Structure And Content Of The Data File

### 3.6.1 Structure Of The Data File

An example data file is shown in Appendix B. This contains, in the commented text, examples of the necessary fields and their values for each data type supported by the metrology application tool.

The fields supported by the data file and their allowed values are discussed below in the following sub-section.

The same conventions and restrictions regarding comment lines, field record specification and grouping are used in the data file as in the action file. The field names are different, but the structure of a field record follows the same general form as described in the previous sub-section.

### 3.6.2 Field Records Supported In The Data File

This subsection lists and describes each field name supported by the data file, its usage, and the allowed values of the field name. The information necessary to completely specify a data set varies according to the type of data. This means that a different subset of the fields are applicable for each data type supported by the metrology application tool. Examples of the fields necessary for each data type are included in the example data file shown in Appendix B.

Azimuth Field Record
Field Name : AZIMUTH =
Field Value : This value is an integer, depending upon the data type of the data for which it is being used, and conforming to the conventions described below.

Description : This field stores the azimuthal angle appropriate to a given data type. Different conventions are adopted for different data types. The uses of the AZIMUTH field are described below. Where a data type is not explicitly mentioned, the user should assume that the AZIMUTH field is not necessary.

The PROMAP MANDREL data retrievals assume that the azimuth is one of four values (0, 90, 180, 270 inclusive) indicating a fixed point around the circle.

The PROMAP SHELL and PSDF data retrievals assume that the azimuth is one of four values (0, 90, 180, 270 inclusive) indicating a fixed point around the circle.

UPMC axial measurements require an azimuth in 10° steps from 0° to 360°.

UPMC azimuthal measurements assume that the azimuth is one of seven values between 1 and 7 inclusive indicating a fixed position along the optical axis at which the measurement was made.

BSDF data is measured in two azimuthal resolutions; from -90° to -20° in 5° steps and from -20° to +20° in 0.25° steps. The AZIMUTH field value should be set to -90 or 20 to indicate which of the resolution files is selected.

Company Name Field Record
Field Name : COMPANY =
Field Value : The following field values are allowed :

ZEISS
MEDALLARIO

Description : The name of the company from which the data originated. This field is essential for PROMAP and PROMAP PSDF data due to different filename formats adopted by the respective manufacturers. ZEISS should be used for all MANDREL data and MEDALLARIO for all shell data files.

Company Name Field Record
Field Name : DATA TYPE =
Field Value : The following field values are allowed :

UPMC_AXIAL
UPMC_AZIMUTHAL
PROMAP_25
PROMAP_20
PROMAP_40
PSDF PROMAP_25 - manufacturer supplied PSDF of PROMAP 2.5 data.
PSDF PROMAP_20 - manufacturer supplied PSDF of PROMAP 20 data.
PSDF PROMAP_40 - manufacturer supplied PSDF of PROMAP 40 data.
SD FOCAL DATA - screening device data focal point parameters of each shell.
SD SERIES COEFFS - screening device data holding the surface error series expansion coefficients used to parametrise the shell shape.
CMD
BSDF - BSDF flux values.

Description : This is a string indicating the type of data to be retrieved from the metrology database. The allowed record values are discussed above.

Data Subset Field Record
Field Name : EXTRACT DATA SETS =
Field Value : The value of the record has the form :

num-1 % num-2 % . . . % num-n

where num-i are integers indicating the identifier of data set to be taken. E.g To take the first and third data sets of PROMAP data, the following record value should be set :

EXTRACT DATA SETS = 1 % 3

Description : This is a field used to extract one or more data sets from a multiple data set series. It is only used for PROMAP and PROMAP PSDF data in which a single file contains up to 10 rows of data measurements.

Flux Incidence Angle Field Record

Field Name : FLUX INCIDENCE =

Field Value : Integer equal to one of the allowed incidence angles discussed below.

Description : BSDF measurements are measured at one of five incident angles (-80°,-83°,-86°,-88°,-89°). This field should be set to one of these values to indicate the appropriate data set.

Mandrel Number Field Record

Field Name : ID MANDREL =

Field Value : Three digit integer. For example, 002.

Description : This is an integer in three digit form indicating the mandrel identifier of the mandrel used to form a mirror shell.

Mirror Model Field Record

Field Name : MIRROR MODEL =

Field Value : Alphanumeric characters (unquoted) indicating a valid mirror model.

Description : This is a string indicating the mirror model containing the shell e.g. FM1, QM etc. Note that the mandrel data is considered to belong to the mirror model MANDREL. Note also that the mirror model represents a directory in the data tree and should be entered in the case appropriate to the directory name.

Measurement Number Field Record

Field Name : NUMBER OF MEASUREMENT =

Field Value : Integer.

Description : Multiple measurements of the same type may be present in the database for a given shell or mandrel. This is an integer indicating the specific measurement number of interest.

Object Type Field Record

Field Name : OBJECT TYPE =

Field Value : The field record can have the following values :

```
MANDREL
SHELL
```

Description : This indicates whether the data of interest is for a mandrel or mirror shell created from a mandrel.

Production Number Field Record

Field Name : PRODUCTION NUMBER =

Field Value : Integer.

Description : Multiple shells are derived from the same mandrel and appear in different mirror models. This field is an integer indicating the unique production number of a specific shell made from a given mandrel.

Mirror Region Field Record
Field Name : REGION =
Field Value : This field record can have the following literal values :

END
MIDDLE
INTERSECTION

Description : This is a string indicating the mirror section region at which the measurement data is taken. It is used for PROMAP and PSDF data.

Mirror Section Type Field Record
Field Name : SECTION TYPE =
Field Value : This field record can have the following literal values :

PARABOLA
HYPERBOLA

Description : This is a string indicating the mirror section type for which measurement data is taken.

Wavelength Field Record
Field Name : WAVELENGTH =
Field Value : This field record takes one of the two values discussed below.

Description : BSDF data is measured at one of two light wavelengths (633 nm and 325 nm). This field should be set to one of these two values to indicate the chosen data set.

# 4 The Mirror Metrology Database

This section discusses the organisation and structure of the mirror metrology database. Information relating to the naming conventions of individual data files in the database, and the internal structure structure of those data files, is discussed in Section 5.

## 4.1 Metrology Data Organisation

The mirror metrology database consists of a hierarchical directory tree containing flat files in which the metrology data is stored. The hierarchical organisation is based around mirror model, mandrel number from which the mirror shell was created, and finally metrology data type, in order of hierarchical ranking from top to bottom.

The mirror model is a project mnemonic used to identify different mirror assembly instances (FM1 or DM, for example). There is no restriction on this name. Note that MANDREL must be included as a mirror model as all mandrel data sets are stored under this directory branch.

The metrology data type is a mnemonic chosen to represent a given data type supported by the metrology application tool.

The mandrel number is represented as a three digit integer, 056 for example.

Below is a symbolic illustration of the directory tree used in the metrology database.

```
/met-db-root
        -> /model-mnemonic-1
                -> /data-type-mnemonic-1
                        -> /shell-id-1
                        -> ...
                        -> /shell-id-n
                -> /data-type-mnemonic-2
                        -> /shell-id-1
                        -> ...
                        -> /shell-id-n
                -> ...
                -> /data-type-mnemonic-i
                        -> /shell-id-1
                        -> ...
                        -> /shell-id-n
        -> /model-mnemonic-2
                -> /data-type-mnemonic-1
                        -> /shell-id-1
                        -> ...
                        -> /shell-id-n
```

```
        -> /data-type-mnemonic-2
                -> /shell-id-1
                -> ...
                -> /shell-id-n
        -> ...
        -> /data-type-mnemonic-i
                -> /shell-id-1
                -> ...
                -> /shell-id-n
-> ...
-> /model-mnemonic-m
        -> ...
```

In this directory tree, the actual data files discussed in Section 5 appear only inside the shell-id-n sub-directories.

It is important to note that each different mirror model may have a different set of mirror shell id sub-directories.

In the current configuration, the metrology database is held in the directory /data/transport/dholling/xmmdata.

## 4.2 Adding Data To The Database

Data is added to the metrology database by copying files delivered by the manufacturer from the external delivery media into the relevant mirror model/data type/shell id subdirectory in the metrology database hierarchy. Occasionally the raw source data files need to be pre-processed to remove formatting anomalies. The pre-processing routines are detailed in Section 6.

## 4.3 Creation And Deletion Of The Database

Unix c-shell scripts exist to create the metrology database directory tree in any directory, and delete all data files and sub-directories in the metrology database directory tree.

The scripts and associated data files are found in the system directory /usr4/users/dholling/xmm/mirrordb/script/mmdb-tree.

Note that the location of the metrology database data is completely independent of the location of the metrology application tool software.

The script to create the metrology database directory tree is called mmdb_dirtree.mak and that to delete the database directory mmdb_dirtree.rem.

The script mmdb_dirtree.mak uses two main source data files to create the database directory tree. These files are named models.dat and datatypes.dat, examples of which are found in the system directory /usr4/users/dholling/xmm/mirrordb/script/mmdb-tree.

The file models.dat contains all mirror model project mnemonics, each mnemonic entered on a single line. For example, MANDREL and FM1 should be included on different lines of this file.

The file datatypes.dat contains the mnemonics chosen for each data type supported by the metrology database. Each mnemonic is entered on a single line in the file. For example, PROMAP and BSDF should be included on different lines of this file.

For each mirror model listed in the file models.dat, there should be a corresponding configuration file named model-mnemonic.dat. This file consists of a number of lines, each line containing a single three digit number indicating the mandrel number of a shell for which metrology data exists in the database for the given model. For example, if metrology data exists for the mirror shell derived from mandrel 57 in mirror model FM1, there should be a corresponding line 057 placed in the file FM1.sid.

Note that there are metrology application constants in the file read.stat that are set, or implicitly contain, the chosen values for the mnemonics (notably in creating the file pathnames of the logically specified data sets in the data configuration file) and care should be taken in changing the contents of the script configuration files.

To run either script, the following procedure should be followed :

1. Copy the script of interest into the top level directory in which the database directory tree is to be created or removed /met-db-root by typing :

unix-prompt> `cp /usr4/users/dholling/xmm/mirrordb/script/script-name /met-db-root`

2. Ensure that the script data files models.dat and datatypes.dat are either created or copied to the top level directory in which the database directory tree is to be created or removed. In addition, ensure that the shell data files *.sid are present for each mirror model listed in the file models.dat.

3. Change directory to the database root directory by typing :

unix-prompt> `cd /met-db-root`

4. Run the script by typing :

unix-prompt> `sh script-name`

where the script name is either mmdb_dirtree.rem or mmdb_dirtree.mak.

## 4.4 Adding A New Data Type To The Metrology Database

A new data type sub-directory needs to be added for each mirror model identified in the top level of the metrology database directory hierarchy. This directory may be created without disturbing the existing metrology database structure and content using the following steps :

1. Edit the script file datatypes.dat to add the new data type mnemonic.

2. Create the shell data file *data-type-mnemonic*.sid for the new data type.

3. Run the create script by typing :

   unix-prompt> `sh mmdb_dirtree.mak`

# 5 Data File Structure In The Metrology Database

This section lists the data sets supported by the metrology application tool, the file naming conventions of these data files in the metrology database, and the format of the contents.

## 5.1 BSDF Reflectivity Data

### 5.1.1 File Structure And Content

The structure, content and naming convention of these data files are identical to that delivered by the manufacturer.

The BSDF data files consist of two line formats. The first line of the file always comprises the n elevation scan range values for which the BSDF values are calculated. n has a value of either 161 (for a flux incidence angle not equal to -89° in which the angles 50° to 90° are covered in 0.25° steps) or 101 (for a flux incidence angle of -89° in which the angles 70° to 90° are covered in 0.2° steps).

The remaining lines each contain n + 1 values. The first value of the line is always the azimuth value in degrees, followed by n values for the range of elevation angles scanned. Each of these n values is the value of the logarithm to base 10 of the ratio of the scattered to incident flux at the given azimuth value.

Shown below is the start and end of the data file 633-86a.txr, which contains the BSDF reflectivity measurements.

```
5e+01 5.025e+01 5.05e+01 5.075e+01 ... 8.925e+01 8.95e+01 8.975e+01 9e+01
-1.999e+01 -5.582e+00 -5.547e+00 ... -8.177e+00 -8.796e+00 -9.426e+00
-1.974e+01 -5.523e+00 -5.566e+00 ... -8.002e+00 -8.629e+00 -9.371e+00

          ⋮

1.951e+01 -5.54e+00 -5.566e+00 ... -8.643e+00 -9.346e+00 -9.699e+00
1.976e+01 -5.537e+00 -5.59e+00 ... -8.721e+00 -9.371e+00 -9.757e+00
2.001e+01 -5.59e+00 -5.586e+00 ... -8.662e+00 -9.371e+00 -9.699e+00
```

### 5.1.2 Filename Convention

The filenames of BSDF data files conform to the following template :

www-nnp.txr

where,

> www represents the wavelength of light used for the measurement. This has the values 633 or 325.
>
> nn represents the angle of flux incidence of the light for the measurement. The allowed values are 80, 83, 86, 88 and 89.
>
> p represents the azimuthal scan range of the measurement. Allowed values are a and b. b represents an azimuth scan range from -90° to -20° in 5° steps. a represents an azimuth scan range from -15° to +15° in 0.1° steps for an incidence angle of -89°, and -20° to +20° in 0.25° steps for all other incidence angles.

For example, BSDF data for a mirror shell derived from mandrel 23 and in mirror model FM1 at a flux incidence angle of -83° using light of wavelength 633 nm over the azimuthal scan range of -20° to +20° is stored in the file :

> /metdb-root/FM1/BSDF/023/633-83a.txr

## 5.2 Screening Device Focal Point Data

### 5.2.1 File Structure And Content

The structure, content and naming convention of these data files is different to that delivered by the manufacturer as the delivered data is run through a pre-processing routine discussed in Section 6.

The screening device focal point data files consist of a series of lines containing the measurement data. Each line of measurement data contains five numbers described below:

> first column = focal plane coordinate [mm]
> second column = X coordinate of the average focus [mm]
> third column = Y coordinate of the average focus [mm]
> fourth column = Half Energy Width [arcsec]
> fifth column = Width of the cone including 90% of the energy [arcsec]

Shown below is the entire data file fm23_2.foc, which contains the focal point data for mirror shell 23.

```
7.49000E+03   2.15750E-01   4.51480E-02   2.27520E+01   5.86980E+01
7.49100E+03   1.80370E-01  -2.25550E-03   2.22320E+01   5.34560E+01
7.49200E+03   1.57360E-01   5.18220E-03   2.14160E+01   5.32470E+01
7.49300E+03   1.77280E-01   2.35370E-02   2.15670E+01   5.40160E+01
7.49400E+03   7.82420E-02   3.23700E-02   2.07850E+01   5.00520E+01
```

| | | | | |
|---|---|---|---|---|
| 7.49500E+03 | 9.07290E-02 | 1.89570E-02 | 2.00030E+01 | 5.19380E+01 |
| 7.49600E+03 | 6.36000E-02 | 4.80360E-03 | 1.93780E+01 | 4.91780E+01 |
| 7.49700E+03 | 8.83440E-02 | 4.42530E-02 | 1.86260E+01 | 5.18370E+01 |
| 7.49800E+03 | 9.48940E-02 | 2.46450E-02 | 1.82910E+01 | 5.09830E+01 |
| 7.49900E+03 | 5.82860E-02 | 3.43300E-02 | 1.80820E+01 | 5.11520E+01 |
| 7.50000E+03 | 7.33570E-02 | -7.42730E-03 | 1.80160E+01 | 5.17710E+01 |
| 7.50100E+03 | 5.97800E-02 | 3.33190E-02 | 1.83860E+01 | 5.13650E+01 |
| 7.50200E+03 | 4.93530E-02 | 2.48020E-02 | 1.79050E+01 | 5.22610E+01 |
| 7.50300E+03 | 4.78590E-02 | 5.61690E-03 | 1.76560E+01 | 5.17590E+01 |
| 7.50400E+03 | 2.94460E-02 | 1.75190E-02 | 1.82620E+01 | 5.43640E+01 |
| 7.50500E+03 | 4.25680E-02 | 2.59370E-02 | 1.93860E+01 | 5.40440E+01 |
| 7.50600E+03 | 3.52670E-02 | -1.29270E-02 | 1.94540E+01 | 5.60600E+01 |
| 7.50700E+03 | 1.51800E-02 | 1.35580E-02 | 1.97620E+01 | 5.60490E+01 |
| 7.50800E+03 | 4.17320E-03 | 1.49260E-02 | 2.08730E+01 | 5.73730E+01 |
| 7.50900E+03 | 1.09220E-02 | 1.76010E-02 | 2.20960E+01 | 5.83500E+01 |
| 7.51000E+03 | -3.84380E-02 | 1.17570E-01 | 2.21670E+01 | 5.78630E+01 |

### 5.2.2 Filename Convention

The filenames of the screening device focal point data files conform to the following template :

fmss_p.foc

where,

ss represents the mandrel number used to form the mirror shell.

p represents the replication number of the shell from the mandrel.

For example, screening device focal point data for a mirror shell derived from mandrel 23 as the second replica, and in mirror model FM1 is stored in the file :

/metdb-root/FM1/SD/023/fm23_2.foc

## 5.3 Screening Device Error Series Expansion Coefficients

### 5.3.1 File Structure And Content

The structure, content and naming convention of these data files is different to that delivered by the manufacturer as the delivered data is run through a pre-processing routine discussed in Section 6.

The screening device focal point data files consist of seven lines containing the coefficients of the orthogonal series expansion. The first four lines contain the cosine error descriptors (in micrometers) and the subsequent three lines contain the sine error descriptors (in micrometers).

The cosine error descriptors are entered in Fourier term order from order 0 to order 3 inclusive (lines 1 to 4), and the sine error descriptors are entered in Fourier term order from order 1 to order 3 inclusive (lines 5 to 7).

Each line of measurement data contains three numbers representing the $0^{th}$, $1^{st}$, and $2^{nd}$ order terms in the Legendre series expansion for a specific order Fourier term. The $0^{th}$ order term is the first column and the $2^{nd}$ order term is the third column.

Shown below is the entire data file fm23_2.ort, which contains the orthogonal surface error series coefficients for mirror shell 23.

```
      0.0000         0.0000         0.0000
     -1.9290        12.5900        -0.0160
     44.3600        30.1600         0.6175
     -9.1090         6.3160        -0.1792
      0.1594        -9.5320         0.1916
    -14.7800         9.4660        -0.8247
      5.7910         6.5580         0.3817
```

For example, the error coefficient of the $1^{s}t$ order Legendre term of the second order Fourier sine term is 9.4660 microns.

### 5.3.2 Filename Convention

The filenames of the screening device error expansion coefficient data files conform to the following template :

fmss_p.ort

where,

ss represents the mandrel number used to form the mirror shell.

p represents the replication number of the shell from the mandrel.

For example, screening device error expansion coefficients for a mirror shell derived from mandrel 23 as the second replica, and in mirror model FM1 is stored in the file :

/metdb-root/FM1/SD/023/fm23_2.ort

### 5.4 PROMAP Shell Data

#### 5.4.1 File Structure And Content

The PROMAP shell data files consists of a ten line header and a number of profile sets containing the profile measurement data for a given row. The number of profile sets is indicated by the ROWS = field of the header, but is usually ten. Each profile set has a row number as the first entry followed by a number of profile measurements equal in number to the value of the COLS = field of the header. The row numbers normally start at 60 and increment by 40 at each profile set up to and including 420. The profile sets are stored in the file in order of increasing row number.

The measurement value in each profile set is in microns but is purely relative, as a parabolic curve is fitted to and then subtracted from the data values after reading the data into the metrology application tool.

Shown below are extracts of a series of profile sets of the data file sf24214a.d10.

```
ZEISS
c:\pro_dat\ms24_2\sf24214a.d10  01.12.95  15:53

XPIXEL=  .31200000E+00
YPIXEL=  .32500000E+00

ZSCALA=  .10000000E+01
COLS=  640
ROWS =  10
BAD_PIXEL_VALUE =   .30000000E+16
ROW#    60
-.16428450E+00
-.16396960E+00
-.16393270E+00
-.16387290E+00
-.16364260E+00
-.16355330E+00

            ⋮

-.16394270E+00
-.16354190E+00
-.16345590E+00
-.16335920E+00
ROW#   100
-.22099650E+00
-.22093410E+00
```

```
-.22079510E+00
-.22073000E+00

              ⋮

-.59562440E+00
-.59528000E+00
ROW#   420
-.64247500E+00
-.64240650E+00
-.64240650E+00
-.64228890E+00
-.64218290E+00

              ⋮
```

### 5.4.2 Filename Convention

The filenames of the Medialario PROMAP shell data files conform to the following template :

sbccdefg.d10

where,

b represents the resolution of the PROMAP measuring instrument. The allowed values are p for resolution x2.5 and f for resolution x40.

cc represents the mandrel number used to form the mirror shell.

d represents replication number of the shell from the mandrel.

e represents the measurement number.

f represents the axial coordinate at which the PROMAP measurent was made. The allowed values are 2 for Z = 7250 mm, 3 for Z = 7350 mm, 4 for Z = 7450 mm, 5 for Z = 7550 mm, 6 for Z = 7650 mm and 7 for Z = 7750 mm. Values 2, 3 and 4 correspond to the end, middle and parabola/hyperbola intersection regions of the parabolic mirror respectively. Values 7, 6 and 5 correspond to the end, middle and parabola/hyperbola intersection regions of the hyperbolic mirror respectively.

g represents the azimuthal angle at which the PROMAP measurement was made. The allowed values are a for angle 0°, b for angle 90°, c for angle 180°, and d for angle 270°.

For example, the third PROMAP x2.5 measurement data set in the middle of the parabolic mirror section at 90° azimuth for a mirror shell derived from mandrel 23 as the second replica, and in mirror model FM1 is stored in the file :

/metdb-root/FM1/PROMAP/023/sp23233b.d10

## 5.5 PROMAP PSDF Shell Data

### 5.5.1 File Structure And Content

The PROMAP PSDF shell data files consists of an eleven line header and a number of profile sets containing the PSDF profile measurement data for a given row. The number of profile sets is indicated by the ROWS = field of the header, but is usually ten. Each profile set has a row number as the first entry followed by a number of profile measurements equal in number to the value of the COLS = field of the header. The row numbers normally start at 60 and increment by 40 at each profile set up to and including 420. The profile sets are stored in the file in order of increasing row number.

The measurement data in each line of a profile set consists of two values. The first value is the reciprocal of the frequency value at which the PSDF is calculated (in mm). The second value represents the PSDF value in TBD.

On reading PSDF data in, the metrology application tool automatically converts the coordinate space value into a frequency value by reciprocating it.

Shown below are extracts of a series of PSDF profile sets of the data file sf24214a.psd.

```
ZEISS
c:\pro_dat\ms24_2\sf24214a.d10  01.12.95  15:53

XPIXEL=  .31200000E+00
YPIXEL=  .32500000E+00

ZSCALA=  .10000000E+01
COLS = 100
ROWS =  10
BAD_PIXEL_VALUE =   .30000000E+16
MAGN=  .40000000E+02
ROW#    60
.199368      .10260130E-06
.099684      .26371260E-07
.066456      .15849540E-07
.049842      .35520330E-07
.039874      .14462120E-08
.033228      .18253950E-07
```

```
.028481       .53949110E-08

                ⋮

.002077       .32352960E-10
.002055       .32581220E-10
.002034       .70683960E-10
.002014       .21349110E-09
.001994       .25829860E-10
ROW#   420
.199368       .42663100E-09
.099684       .47766680E-08
.066456       .27525620E-08
.049842       .95785970E-09
.039874       .61167000E-08
.033228       .61640910E-08
.028481       .11834770E-08
.024921       .12315420E-09
.022152       .21287430E-08
.019937       .15432250E-08
.018124       .15515870E-08
.016614       .58114940E-08
```

### 5.5.2 Filename Convention

The filenames of the Medialario PROMAP shell PSDF data files conform to the following template :

sbccdefg.psd

where,

b represents the resolution of the PROMAP measuring instrument. The allowed values are p for resolution x2.5 and f for resolution x40.

cc represents the mandrel number used to form the mirror shell.

d represents replication number of the shell from the mandrel.

e represents the measurement number.

f represents the axial coordinate at which the PROMAP measurent was made. The allowed values are 2 for Z = 7250 mm, 3 for Z = 7350 mm, 4 for Z = 7450 mm, 5 for Z = 7550 mm, 6 for Z = 7650 mm and 7 for Z = 7750 mm. Values 2, 3 and 4 correspond to the end, middle and parabola/hyperbola intersection regions of the parabolic mirror respectively. Values 7, 6 and 5 correspond to the end, middle and parabola/hyperbola intersection regions of the hyperbolic mirror respectively.

g represents the azimuthal angle at which the PROMAP measurement was made. The allowed values are a for angle 0°, b for angle 90°, c for angle 180°, and d for angle 270°.

For example, the third PROMAP x40 PSDF measurement data set in the middle of the hyperbolic mirror section at 270° azimuth for a mirror shell derived from mandrel 23 as the second replica, and in mirror model FM1 is stored in the file :

/metdb-root/FM1/PROMAP/023/sf23236d.psd

## 5.6 PROMAP Mandrel Data

### 5.6.1 File Structure And Content

The PROMAP mandrel data files consists of a ten line header and one profile set containing the profile measurement data for a given row. The number of profile sets is indicated by the ROWS = field of the header, but is usually one. Each profile set has a text string as the first entry followed by a number of profile measurements equal in number to the value of the COLS = field of the header.

The measurement value in each profile set is in microns but is purely relative, as a parabolic curve is fitted to and then subtracted from the data values after reading the data into the metrology application tool.

Shown below are extracts of a series of profile sets of the data file 24p18m0.asc.

```
VER = DCON1

TYPE_OF_DATA = RAW
XPIXEL = 4.965635e+00
YPIXEL = 5.158730e+00
DATA_TYPE = MICRONS
ZSCALE = 1.000000e+00
COLS = 640
ROWS = 1
BAD_PIXEL_VALUE = 3.000000e+15
DATA_VALUES
```

```
3.880372e+00
3.880298e+00
3.880085e+00
3.879951e+00
3.879993e+00
3.880252e+00
3.880442e+00
3.880601e+00
3.880507e+00
3.880432e+00
3.880578e+00

        ⋮

3.892199e+00
3.892452e+00
3.892385e+00
3.892479e+00
3.892563e+00
3.892555e+00
3.892387e+00
3.892266e+00
3.892352e+00
3.892341e+00
3.892235e+00
```

### 5.6.2 Filename Convention

The filenames of the Carl-Zeiss PROMAP mandrel data files conform to the following template :

bsstdde0.asc

where,

b represents the resolution of the PROMAP measuring instrument. The allowed values are ' ' (i.e. no character) for resolution x2.5 and x for resolution x20.

ss represents the mandrel number.

t represents the mirror section measured. The allowed values are h for the hyperbolic mirror section and p for the parabolic mirror.

dd represents the azimuthal angle at which the PROMAP measurement was made. The allowed values are 00 for angle 0°, 90 for angle 90°, 18 for angle 180°, and 27 for angle 270°.

e represents the axial coordinate at which the PROMAP measurent was made. The allowed values are r, m, and k for the end, middle and parabola/hyperbola intersection regions of the mirror respectively.

For example, the PROMAP x2.5 measurement data set in the middle of the hyperbolic mirror section at 270° azimuth for mandrel 23 is stored in the file :

/metdb-root/MANDREL/PROMAP/023/23h27m0.asc

## 5.7 UPMC Axial Profile Shell And Mandrel Data

### 5.7.1 File Structure And Content

The UPMC axial profile data files consists of a series of lines containing the profile measurement data. Each line of measurement data contains two numbers, the first is the axial coordinate in mm at which the measurement is made and the second is the radial distance of the mandrel or shell surface from the axial axis in mm. Axial measurements are taken at roughly 1 mm separation from the end of the parabolic section (-560 mm), where the light enters the mirror assembly, until the end of the hyperbolic section (0 mm), where the light leaves the mirror assembly.

Shown below is the start and end of the data file m120.asc, which contains the axial profile data for mirror mandrel 1 at an azimuthal angle of 120 degrees.

```
-560.8611      349.6512
-559.86        349.6397
-558.8606      349.6282
-557.8643      349.6171
-556.8652      349.6057
-555.8605      349.5938
-554.8611      349.5818
-553.8632      349.5703

                 ⋮

-18.0414       336.6052
-17.0426       336.5709
-16.0415       336.5362
-15.04         336.5013
-14.0411       336.4665
```

```
-13.045         336.4327
-12.044         336.3988
-11.0416        336.3651
-10.0354        336.332
-9.1059         336.3013
-8.0312         336.2645
-7.0311         336.2325
-6.033          336.1988
```

### 5.7.2 Filename Convention

The filenames of the UPMC axial profile mandrel data files conform to the following template :

tzzz.sss

where,

> t represents the object for which the UPMC axial profile measurements are made. The allowed values are m for mandrels and s for mirror shells.
>
> zzz represents the azimuthal angle at which the UPMC measurement was made. The allowed values are 000 to 350 for azimuthal angles from 0° to 350° respectively. The azimuthal angle increments in 10° steps.
>
> sss represents the mirror section measured. The allowed values are hyp for the hyperbolic mirror section and par for the parabolic mirror.

For example, the UPMC axial measurement data set of the hyperbolic mirror section for mandrel 24 at an azimuthal angle of 170° is stored in the file :

/metdb-root/MANDREL/UPMC/024/m170.hyp

## 5.8 UPMC Azimuthal Profile Mandrel Data

### 5.8.1 File Structure And Content

The UPMC azimuthal profile data files consists of four lines of header information followed by a series of lines containing the profile measurement data. Each line of measurement data contains two numbers, the first is the azimuthal angle in degrees at which the measurement is made and the second is the radial distance of the mandrel surface from the axial axis in mm. Azimuthal measurements are taken at roughly 1 degree separation from approximately 0 to 360 degrees.

Shown below is the start and end of the data file kreis-2.asc, which contains the azimuthal profile data for mirror mandrel 1.

```
Mirror Shell Label:            XMM.01.073.MDM2
Date of measurement:           19.04.1995
Circumferential profile at Z = -49
Mirror Shell Number:            1
 6.607099E-04  338.2022
 .9687567      338.1875
 1.973035      338.167
 2.977311      338.1466
 3.974933      338.126
 4.951344      338.1044
 5.983618      338.0805
 6.946732      338.0586
 7.952042      338.036

             ⋮

 352.1304      338.3187
 353.1012      338.3065
 354.1112      338.2914
 355.0878      338.2784
 356.1226      338.263
 357.072       338.2489
 358.0818      338.2325
```

### 5.8.2 Filename Convention

The filenames of the UPMC azimuthal profile mandrel data files conform to the following template :

kreis-n.asc

where,

> n represents the axial coordinate at which the UPMC azimuthal measurement was made. The allowed values are 1 to 7 for axial coordinates spaced by approximately 100 mm from the light entry point to the mirror assembly to the exit point, respectively.

For example, the UPMC azimuthal measurement data set of the end of the hyperbolic mirror section for mandrel 24 is stored in the file :

/metdb-root/MANDREL/UPMC/024/kreis-7.asc

## 5.9 CMD Mandrel Data

### 5.9.1 File Structure And Content

The CMD profile data files consists of eight lines of header information followed by a series of lines containing data whose meaning is **TBD**.

Shown below is the start and end of the data file h24105, which contains the CMD profile data for the hyperbolic section of mirror mandrel 24 at at 225° azimuth.

```
H24
27.01.94
14:27
ABSCHLUSSMESSUNG VOR MIKROPOLITUR
H24105
105,225,7180
4000,.00271451,.08
0,0,0,0
-8192,8188,8188,8188,8188,8188,6268,4212,2960,2280,1952,1804
1724,1660,1608,1572,1532,1508,1484,1464,1436,1424,1400,1388
1376,1364,1344,1332,1324,1308,1300,1292,1284,1276,1264,1256
1240,1220,1220,1236,1208,1220,1180,1164,1180,1148,1184,1152
1180,1132,1164,1112,1144,1108,1132,1108,1120,1100,1104,1084
1092,1084,1088,1080,1072,1068,1060,1060,1056,1048,1044,1040
1032,1036,1024,1024,1016,1020,1008,1004,1000,996,996,992
984,988,980,976,972,972,964,960,960,956,948,948
944,944,936,936,932,928,924,924,916,912,908,916

        ⋮

212,220,224,228,236,244,252,260,260,268,280,280
288,296,304,308,320,328,336,340,348,356,360,364
380,388,392,404,408,420,432,444,452,468,472,488
500,516,524,540,552,564,576,596,604,624,644,668
688,716,744,780,820,876,940,1020,1128,1264,1436,1640
1880,2156,2484,2840
```

### 5.9.2 Filename Convention

The filenames of the CMD mandrel data files conform to the following template :

bss10n

where,

> b represents the mirror section measured. The allowed values are h for the hyperbolic mirror section and p for the parabolic mirror.
>
> ss represents the mandrel number.
>
> n represents the azimuthal angle at which the CMD measurement was made. The allowed values are 0 to 7 for azimuthal angles spaced by 45° starting from 0° up to 315°, respectively.

For example, the CMD measurement data set of the hyperbolic mirror section at 225° azimuth for mandrel 24 is stored in the file :

> /metdb-root/MANDREL/CMD/024/h24105

# 6 Pre-Processing Of Raw Data Files

There has been an inconsistent policy regarding data deliverables by the manufacturers to ESTEC during the early part of the project. This has resulted in inconsistent file formats for the same type of data delivered by different manufacturers and file contents that do not match the documented file formats (if available).

As a result of this, several routines have been written, and are available in the metrology application tool, to pre-process the raw data delivered by the manufacturers into a form more amenable to the data read routines inside the metrology application tool.

The sub-sections below detail which pre-processing routines should be run inside the IDL environment on which raw data sets before that data is included in the metrology database.

Note that once the metrology application tool has been compiled inside IDL, each of the routines below are available for calling either from the command line or from other IDL procedures.

## 6.1 Shell PROMAP PSDF Files

Raw PROMAP PSDF data files are pre-processed using the function :

redCLEAN_MEDALLARIO_PSDF

This function takes two arguments :

**arg1** a string containing the file pathname of the input file to be pre-processed.

**arg2** a string containing the file pathname of the pre-processed, output file.

Note that the output filename should follow the convention described in Section 5. The raw input files also correspond to this naming convention, so care must be taken to ensure that either the input files are renamed to a different name if the output file is located in the same directory as the input file, or the input file is placed in a different directory to that specified in the output file pathname.

For example, if a file sf23213a.psd is to be pre-processed, either one of the following two function calls may be used in IDL to pre-process the data.

```
idl-prompt> ret-value = redCLEAN_MEDALLARIO_PSDF(
'/source-directory/sf23213a.psd',
'/target-directory/sf23213a.psd')
```

if the source-directory pathname is different to the target-directory pathname, or, for the same source and target directories,

```
idl-prompt> ret-value = redCLEAN_MEDALLARIO_PSDF(
'/data-directory/sf23213a.pre',
'/data-directory/sf23213a.psd')
```

where the source file has been renamed to sf23213a.pre before pre-processing.

### 6.2 Screening Device Text Files

Raw screening device data files are pre-processed using the function :

redPREPROCESS_SCREENING

This function takes two arguments :

arg1 a string containing the file pathname of the input file to be pre-processed.

arg2 a string containing the file pathname **up to the filename stem** of the pre-processed, output file.

Note that the output filename stem should follow the convention described in Section 5.

The filename stem (i.e. without suffix) of the raw input files corresponds to this naming convention, but with the suffix of .txt. The raw files contain both focal point data and the orthonormal error series expansion coefficient data for a given mirror shell. After pre-processing, the raw data is split into two separate files following the conventions in Section 5.

For example, if a file fm23_2.txt is to be pre-processed, the following function call may be used in IDL to pre-process the data.

```
idl-prompt> ret-value = redPREPROCESS_SCREENING(
'/source-directory/fm23_2.txt', '/target-directory/fm23_2')
```

Note that there are no restrictions on the source-directory pathname and the target-directory pathname.

### 6.3 Shell PROMAP Files

Raw shell PROMAP data files from Medialario are pre-processed using the function :

redCOMB_MEDALLARIO_PROMAP

This function takes one argument :

arg1 a string containing the file pathname **up to the filename stem** of the input files to be pre-processed.

This routine takes ten shell PROMAP row measurement files, labelled *.a01 ... *.a10, and combines them into a single, ten row, file labelled *.d10, following the conventions in Section 5.

For example, if the files sp23212a.a01 ... sp23212a.a10 are to be pre-processed, the following function call may be used in IDL to pre-process the data.

```
idl-prompt> ret-value = redCOMB_MEDALLARIO_PROMAP(
'/source-directory/sp23212a')
```

Note that there is no restriction on the source-directory pathname.

## 6.4 Mandrel PROMAP Files

Raw PROMAP mandrel data files are pre-processed using the function :

redCZ_TO_PROMAP_25

This function takes two arguments :

arg1 a string containing the file pathname of the input file to be pre-processed.

arg2 a string containing the file pathname of the pre-processed, output file.

Note that the output filename should follow the convention described in Section 5. The raw input files also correspond to this naming convention, so care must be taken to ensure that either the input files are renamed to a different name if the output file is located in the same directory as the input file, or the input file is placed in a different directory to that specified in the output file pathname.

For example, if a file 40p00m0.asc is to be pre-processed, either one of the following two function calls may be used in IDL to pre-process the data.

```
idl-prompt> ret-value = redCZ_TO_PROMAP_25(
'/source-directory/40p00m0.asc',
'/target-directory/40p00m0.asc')
```

if the source-directory pathname is different to the target-directory pathname, or, for the same source and target directories,

```
idl-prompt> ret-value = redCZ_TO_PROMAP_25(
'/data-directory/40p00m0.pre', '/data-directory/40p00m0.asc')
```

where the source file has been renamed to 40p00m0.pre before pre-processing.

# A Example Action File

```
#
# XMM MIRROR METROLOGY DATABASE
#
# Author : D. M. Holling
# Version : 1.0
# Date : 1/3/96
# Type : Action definition file
#
#----------------------------------------------------------------
#
# This file is used to specify an action list on which
# the metrology database top level routine operates.
#
# A number of actions are specified on the data sequentially
# using action records. The action types and record
# structures are defined below.
#
# Each action record is specified on a number of lines,
# maximum width of 132 characters, and separated by
# the action record separator.
#
# # :
# This is a comment character. All other text on the
# line is ignored.
#
# - :
# This is an action record separator
#
# % :
# This is an action record field separator
#
# ACTION = :
# This specifies the type of action to be
# performed on the current action record.
# The values of this field supported are :
#
# DISPLAY - display a data set
# FRACTAL FIT - fit a best fit fractal function
# to a data set. Generates/returns
# a 1-D array of two parameters [K,n]
# where K and n are fitted to the data
# set satisfying :
#
#    Kx**n
#
```

```
# x is the independent variable value
# and ** is the power operand
#
# COMBINE - combine regions of two data sets
# into one
# SAVE - save data sets into a specified file.
# The internal database format is
# used which can only be read using the
# RETRIEVE action
# RETRIEVE - retrieve saved data sets from a
# specified file
# EXT AVERAGE - average one or more data sets
# GENERATE SURFACE - generate error term about
# ideal mirror surface for a given order
# of Fourier and Legendre expansion
# term. This MUST use orthogonal error series
# data file as source data.
# GRID SERIES - generate a 'grid' of data sets. It
# is assumed that each set gridded
# has a common independent variable
# value and number of points in the set.
# The sets specified in the SOURCE IDS
# field are taken in order and concatenated
# into a single array. If a contributing
# data set has more than one series in it
# then all series are gridded in order of
# storage in the data set.
# GENERATE DATA SERIES - this action can be
# used to generate a series of points
# satisfying a given function. The data
# series provides equally spaced points
# over a user specified range of values
# of independent variable (x-axis).
#
# Actions to be supported in the future are:
#
# PSDF - return PSDF of one or more data sets
#
# CONDITIONS = :
# This specifies the two condition strings
# applied to the data sets being COMBINED,
# each condition being read as a string and
# separated by the '%' character.
# Note that if a data set contains multiple
# dependent variable series, only the first
# series is combined. Note also that the first
# data set specified in the SOURCE IDS list
# takes the first condition string in the
```

```
# CONDITIONS list. A condition string may be
# blank.
#
# Valid conditions are :
# '>= value'
# '<= value'
# '<= value1, >= value2' where value1 >
# value2
#
# Note that the conditions are applied to the
# independent variable.
#
# DATA SERIES TYPE = :
# This specifies the type of data series generated
# by the action GENERATE DATA SERIES. Values
# currently supported are :
#
# POWER - fits equally spaced points on a LOG10
# scale using the formula Kx**n, where K and n
# are supplied by the SOURCE DATA ID and x is
# the independent variable value. Note that K
# is the first array element and n is the second
# array element in  SOURCE DATA ID.
#
# Values to be supported are :
# LINEAR - do a linear fit to the data
#
# EXPANSION = :
# This specifies the term required in the surface
# error calculation. It has four fields in the
# form :
# symmetry term % legendre order % fourier order % radius
#
# where :
# Legendre order = order of Legendre polynomial of
# value 0 <= n <= 2
# Fourier order = order of Fourier coefficient of
# value 0 <= m <= 3
# symmetry term = 'c' or 's' indicating cosine or
# sine part of the Fourier expansion
# respectively.
# radius = radius (in mm) of the cylinder against
# which the surface error calculations
# are plotted.
#
# i.e. THIS FIELD IS ONLY USED BY THE
# 'GENERATE SURFACE' ACTION.
#
```

```
# FILE PATHNAME = :
# This field is interpreted as a text string used
# to specify the full pathname of the file used
# to RETRIEVE or SAVE data in the internal
# database format.
#
# GRAPH TYPE = :
# This specifies the type of graph on which
# to display the data for a graphical display.
# The allowed values are :
#
# LINEAR - plot a linear-linear (x/y) axis graph
# LINEAR-LOG - plot a linear-log (x/y) axis graph
# LINEAR-LOG - plot a log-linear (x/y) axis graph
# LOG-LOG - plot a log-log (x/y) axis graph
#
# LABEL = :
# This field is interpreted as a text string. It
# is used to provide an array of labels for curves
# to be plotted on a single graphical display.
# Consequently there may be more than one label
# field present in an action record definition.
#
# Note that it is the users responsibility to
# define an equivalent number of labels for the
# individual curves to be plotted (remember that
# some data sets contain multiple curves which
# each require an individual label). In addition
# the labels are applied in strict order i.e.
# first label to first curve etc.
#
# OUTPUT TYPE = :
# This specifies the mode of output for a
# display action. The allowed values are :
#
# GRAPH - plot data on a graphical display
# SURFACE - plot data as a 3-D surface
#
# POINT RANGE = :
# This is a two record field specifying the
# lower and upper limit of independent variable
# used for the action GENERATE DATA SERIES. It
# has the form :
# lower value % upper value
#
# RESULT ID = :
# Integer value giving id of data resulting from
# the action
```

```
#
# SOURCE IDS = :
# This specifies the identifiers of the data
# in the data list used as source data for the
# action. Each data set specified in the config
# file is given a monotonically increasing
# integer identifier starting from 1 in the
# data list. Subsequent data sets derived by
# actions are also given integer identifiers.
#
# This record may consist of one or more integer
# identifiers separated by the '%' character.
#
# TITLE = :
# This field is interpreted as a text string used
# as the title of a graphical and surface display.
#
# XTITLE = :
# This field is interpreted as a text string used
# as the x-axis label of a graphical and
# surface display.
#
# YTITLE = :
# This field is interpreted as a text string used
# as the y-axis label of a graphical and
# surface display.
#
# ZTITLE = :
# This field is interpreted as a text string used
# as the z-axis label of a surface display.
#

#----------------------------------------------------------
#
# EXAMPLE ACTION TO SAVE DATA READ IN
#
#----------------------------------------------------------

#ACTION = SAVE
#SOURCE IDS = 1
#FILE PATHNAME = /usr4/users/dholling/xmm/mirrordb/dev/d4.tst

#-

#----------------------------------------------------------
#
# EXAMPLE ACTION TO RETRIEVE DATA SAVED
#
```

```
#----------------------------------------------------------

#ACTION = RETRIEVE
#RESULT ID = 2
#FILE PATHNAME = /usr4/users/dholling/xmm/mirrordb/dev/d4.tst

#-

#----------------------------------------------------------
#
# EXAMPLE ACTION TO GRID DATA SERIES.
#
# All data stored in identifiers 1, 3 and 5
# is gridded into a single array. It is assumed
# that the data in all arrays has measurements
# at all the same independent variable values.
# Each data series in 1 form the first series in
# the output set, 10, followed by the series in
# 3 and then by those in 5.
#
#
#----------------------------------------------------------

#ACTION = GRID SERIES
#SOURCE IDS = 1 % 3 % 5
#RESULT ID = 10

#-

#----------------------------------------------------------
#
# EXAMPLE ACTION TO GENERATE A POWER DATA SERIES.
# The points generated are equally spaced between
# 1.0 and 1.0E6 on the independent variable LOG10
# axes.
#
#----------------------------------------------------------

#ACTION = GENERATE DATA SERIES
#DATA SERIES TYPE = POWER
#SOURCE IDS = 1
#RESULT ID = 10
#POINT RANGE = 1.0 % 1.0E6

#-

#----------------------------------------------------------
#
```

```
# EXAMPLE ACTION TO GENERATE A SURFACE ERROR
# TERM ARRAY OF POINTS. THIS SPECIFIES THE
# COSINE PART OF THE 1st ORDER LEGENDRE AND
# 2nd ORDER FOURIER COMPONENT. THE SURFACE
# DEFORMATIONS ARE GENERATED RELATIVE TO A
# CYLINDER OF RADIUS 0.05mm
#
#----------------------------------------------------------

#ACTION = GENERATE SURFACE
#SOURCE IDS = 1
#RESULT ID = 2
#EXPANSION = c % 1 % 2 % 5E-2

#-

#----------------------------------------------------------
#
# EXAMPLE ACTION TO DISPLAY A SURFACE PLOT
#
# This assumes as input that the source array is
# 3D having the following elements :
# arrOut(i,j,k) - arrOut(*,j,0) contains the
# x coordinate values for z coordinate j
# arrOut(*,j,1) contains the
# y coordinate values for z coordinate j;
# arrOut(*,j,2) contains the z coordinate
# value for axial coordinate j i.e. constant
#
#
#----------------------------------------------------------

#ACTION = DISPLAY
#OUTPUT TYPE = SURFACE
#SOURCE IDS = 1
#TITLE = THIS IS A TEST SURFACE
#XTITLE = This Is the X-axis Label
#YTITLE = This Is the Y-axis Label
#ZTITLE = This Is the Z-axis Label

#-

#----------------------------------------------------------
#
# EXAMPLE ACTION TO FRACTAL FIT A DATA SERIES
#
#
#----------------------------------------------------------
```

```
#ACTION = FRACTAL FIT
#SOURCE IDS = 1
#RESULT ID = 3

#-

#---------------------------------------------------------
#
# EXAMPLE ACTION TO COMBINE TWO DATA SERIES
#
# note that all points having an independent
# variable value <= 0.05 are included from the
# data identified by the identifier 1 and
# all points having an independent
# variable value <= 0.46 are included from the
# data identified by the identifier 2
#
#---------------------------------------------------------

#ACTION = COMBINE
#SOURCE IDS = 1 % 2
#CONDITIONS = <=0.05 % <=0.46
#RESULT ID = 3

#-

#---------------------------------------------------------
#
# EXAMPLE ACTION TO AVERAGE TWO DATA SERIES
#
# Note : all data series in data set 1 are added
# to those in 2 and averaged. The result is
# saved as data set 3, the independent varaible
# values being taken as those from set 1. It is
# assumed that there are an equal number of points
# in set 1 and 2.
#
#---------------------------------------------------------

#ACTION = EXT AVERAGE
#SOURCE IDS = 1 % 2
#RESULT ID = 3

#-

#---------------------------------------------------------
#
```

```
# EXAMPLE ACTION TO DISPLAY A DATA SERIES
#
# Note : all data series in data set 1 are plotted
# on a LOG-LOG graph. The first two data series are
# labelled using the two label strings in the graph
# key. Subsequent data series are unlabelled.
#
#----------------------------------------------------------

#ACTION = DISPLAY
#OUTPUT TYPE = GRAPH
#GRAPH TYPE = LOG-LOG
#SOURCE IDS = 1
#LABEL = Original data
#LABEL = Saved and retrieved data
#TITLE = THIS IS A TEST TITLE
#XTITLE = This Is the X-axis Label
#YTITLE = This Is the Y-axis Label

#-

#----------------------------------------------------------
#
# EXAMPLE ACTION TO DISPLAY TWO DATA SERIES
#
# Note : all data series in data set 1 and 2 are plotted
# on a LOG-LOG graph. The first two data series are
# labelled using the two label strings in the graph
# key. Subsequent data series are unlabelled.
#
#----------------------------------------------------------

#ACTION = DISPLAY
#OUTPUT TYPE = GRAPH
#GRAPH TYPE = LOG-LOG
#SOURCE IDS = 1 % 2
#LABEL = Original data
#LABEL = Saved and retrieved data
#TITLE = THIS IS A TEST TITLE2
#XTITLE = This Is the X-axis Label
#YTITLE = This Is the Y-axis Label

#----------------------------------------------------------
#
# TEST ACTION CHAIN
#
# 1. This chain displays the data sets in the source
# data 1 on a graphical display
```

```
#
# 2. It averages the fields in the data set 2, the
# result being stored in data set 3.
#
# 3. Data set 3 is displayed on a separate graphical
# display
#
# 4. A fractal fit is made to data set 3 and is stored
# as data set 4
#
# 5. Data set 4 is stored in the file
# data-root/data-set4.tmp
#
#-------------------------------------------------------------

#ACTION = DISPLAY
#OUTPUT TYPE = GRAPH
#GRAPH TYPE = LOG-LOG
#SOURCE IDS = 1
#LABEL = PROMAP 40 PSDF ROW#60
#LABEL = PROMAP 40 PSDF ROW#100
#LABEL = PROMAP 40 PSDF ROW#140
#LABEL = PROMAP 40 PSDF ROW#180
#LABEL = PROMAP 40 PSDF ROW#220
#LABEL = PROMAP 40 PSDF ROW#260
#LABEL = PROMAP 40 PSDF ROW#300
#LABEL = PROMAP 40 PSDF ROW#340
#LABEL = PROMAP 40 PSDF ROW#380
#LABEL = PROMAP 40 PSDF ROW#420
#TITLE = THIS IS DATA SET 1 RAW PSDF DATA
#XTITLE = Frequency (in mm**-1)
#YTITLE = PSDF (in units TBD)

#-

#ACTION = EXT AVERAGE
#SOURCE IDS = 2
#RESULT ID = 3

#-

# there should only be one data series to be plotted
# i.e. generated by the average

#ACTION = DISPLAY
#OUTPUT TYPE = GRAPH
#GRAPH TYPE = LOG-LOG
#SOURCE IDS = 3
```

```
#LABEL = Average of PROMAP 2.5 PSDF Data
#TITLE = THIS IS DATA SET 2 AVERAGED
#XTITLE = Frequency (in mm**-1)
#YTITLE = PSDF (in units TBD)

#-

#ACTION = FRACTAL FIT
#SOURCE IDS = 3
#RESULT ID = 4

#-

# save the y-intercept and gradient of the fit

#ACTION = SAVE
#SOURCE IDS = 4
#FILE PATHNAME = data-set4.tmp
```

# B Example Data Definition File

```
#
# XMM MIRROR METROLOGY DATABASE
#
# Author : D. M. Holling
# Version : 1.0
# Date : 1/3/96
# Type : Data configuration definition file
#
#---------------------------------------------------------------
#
# This is a standard data configuration file for the top level
# routine used to process mirror metrology
# data. The following special fields are processed :
#
# 1. Any line beginning with a '#' character are disregarded
# 2. Any leading or trailing whitespace is stripped out.
# 3. Any repetition of a field causes the last value of
#    that field to be used, disregarding all previous
#    values.
# 4. All fields must be exactly as appearing in the list
#    below (including spaces)
# 5. Collections of the fields listed below constitute a
#    GROUP of fields. It is possible to specify multiple
#    groups of fields within the file, each group separated
#    using the group separator '-' as the first non blank
#    character on a line.
# 6. The default number of groups is taken as one.
#
# The fields possible within any group are listed below. There
# may be 0 to n instances of each field within a single group,
# the last instance of a given field being taken as the
# appropriate value as discussed in point 3 above.
#
# # :
# This is a comment character. All other text on the
# line is ignored.
#
# - :
# This is a group separator. All fields occuring after
# this character are ascribed to a new group until the
# next group separator is found or the file ends.
#
#
# AZIMUTH = :
# This should be an integer storing the azimuth
```

```
# of interest.
#
# Note that the PROMAP MANDREL data retrievals
# assume that the azimuth is one of four values
# (0, 90, 180, 270 inclusive) indicating a fixed
# point around the circle.
#
# The PROMAP SHELL and PSDF data retrievals
# assume that the azimuth is one of four values
# (0, 90, 180, 270 inclusive) indicating a fixed
# point around the circle.
#
# UPMC axial measurements require an azimuth in 10
# degree steps from 0 to 360 degrees.
#
# UPMC azimuthal measurements assume that the
# azimuth is one of seven values
# between 1 and 7 inclusive indicating a fixed
# position along the optical axis at which the
# measurement was made.
#
# BSDF data is measured in two azimuthal resolutions;
# from -90 to -20 in 5 degree steps and from
# -20 to +20 degrees in 0.25 degree steps. The
# AZIMUTH field value should be set to -90 or 20
# to indicate which of the resolution files is
# selected.
#
# COMPANY = :
# The name of the company from which the data
# originated. The following values are allowed :
#
# ZEISS
# MEDALLARIO
#
# ZEISS should be used for all MANDREL data and
# MEDALLARIO for all shell data files. This field
# is necessary for PROMAP and PROMAP PSDF data.
#
# DATA TYPE = :
# This is a string indicating the type of data
# to be retrieved. It can have the following
# literal values :
#
# UPMC_AXIAL
# UPMC_AZIMUTHAL
# PROMAP_25
# PROMAP_20
```

```
# PROMAP_40
# PSDF PROMAP_25 - manufacturer supplied
# PSDF of PROMAP 2.5 data
# PSDF PROMAP_20 - manufacturer supplied
# PSDF of PROMAP 20 data
# PSDF PROMAP_40 - manufacturer supplied
# PSDF of PROMAP 40 data
# SD FOCAL DATA - screening data focal
# data parameters of each shell
# SD SERIES COEFFS - screening data holding
# the Legendre series expansion
# coefficients used to parametrise
# the shell shape
# CMD
# PANTER - not used currently
# BSDF - BSDF flux values
#
# EXTRACT DATA SETS = :
# This is a field used to extract one or more
# data sets from a multiple data set series. It
# is only used for PROMAP and PROMAP PSDF data
# in which a single file contains up to 10
# rows of data measurements. The value of the
# record has the form :
# num 1 % num 2 % ...
# where num i are integers indicating the data
# set to be taken. E.g To tkae the first and
# third data sets of PROMAP data, the following
# should be set :
# EXTRACT DATA SETS = 1 % 3
#
# FLUX INCIDENCE = :
# BSDF measurements are measured at one of five
# incident angles (-80,-83,-86,-88,-89 degrees).
# This field should be set to one of these values
# to indicate the appropriate data set.
#
# ID MANDREL = :
# This is an integer in three digit form e.g.
# 002 indicating the source mandrel id.
#
# MIRROR MODEL = :
# This is a string indicating the mirror model
# containing the shell e.g. FM1, QM etc. Note
# that the mandrel data is considered to belong
# to the mirror model MANDREL. Note also that
# the mirror model represents a directory in
# the data tree and should be entered in the
```

```
# case appropriate to the directory name.
#
# NUMBER OF MEASUREMENT = :
# Multiple measurements of the same type may
# be present in the database. This is an integer
# indicating the measurement number of interest.
#
# OBJECT TYPE = :
# This indicates whether the data of interest
# is for a mandrel or mirror shell created
# from a mandrel. It can have the following
# values :
#
# MANDREL
# SHELL
#
# PRODUCTION NUMBER = :
# Multiple shells are derived from the same
# mandrel and appear in different mirror models.
# This field is an integer indicating the unique
# production number of a specific shell made from
# a given mandrel.
#
# REGION = :
# This is a string indicating the mirror section
# region to be measured. It can have the following
# literal values :
#
# END
# MIDDLE
# INTERSECTION
#
# SECTION TYPE = :
# This is a string indicating the mirror section
# to be measured. It can have the following
# literal values :
#
# PARABOLA
# HYPERBOLA
#
# WAVELENGTH = :
# BSDF data is measured at one of two
# light wavelengths (633 nm and 325 nm).
# This field should be set to one of these two
# values to indicate the chosen data set.
#

#-----------------------------------------------------
```

```
#
# BSDF data
#
# This corresponds to the file :
# data-root/FM1/BSDF/023/633-83a.txr
#
#---------------------------------------------------------

#ID MANDREL =    023
#MIRROR MODEL = FM1
#DATA TYPE = BSDF
#WAVELENGTH = 633
#FLUX INCIDENCE = -83
#AZIMUTH = 20

#---------------------------------------------------------
#
# SCREENING DATA FOCAL PARAMETER
# FIT
#
# This corresponds to the file :
# data-root/FM1/SD/023/fm23_2.foc
#
#---------------------------------------------------------

#ID MANDREL =    023
#MIRROR MODEL = FM1
#DATA TYPE = SD FOCAL DATA
#OBJECT TYPE = SHELL
#PRODUCTION NUMBER = 2

#---------------------------------------------------------
#
# SCREENING DATA ERROR COEFFICIENT
# FIT
#
# This corresponds to the file :
# data-root/FM1/SD/023/fm23_2.ort
#
#---------------------------------------------------------

#ID MANDREL =    023
#MIRROR MODEL = FM1
#DATA TYPE = SD SERIES COEFFS
#OBJECT TYPE = SHELL
#PRODUCTION NUMBER = 2

#---------------------------------------------------------
```

```
#
# PROMAP 2.5 SHELL DATA
#
# This corresponds to the file :
# data-root/FM1/PROMAP/023/sp23213a.d10
#
#----------------------------------------------------------

#AZIMUTH =    000
#ID MANDREL =    023
#MIRROR MODEL = FM1
#OBJECT TYPE = SHELL
#DATA TYPE = PROMAP_25
#SECTION TYPE = PARABOLA
#COMPANY = MEDALLARIO
#REGION = MIDDLE
#NUMBER OF MEASUREMENT = 1
#PRODUCTION NUMBER = 2

#----------------------------------------------------------
#
# PROMAP 40 SHELL DATA
#
# This corresponds to the file :
# data-root/FM1/PROMAP/023/sf23213a.d10
#
#----------------------------------------------------------

#AZIMUTH =    000
#ID MANDREL =    023
#MIRROR MODEL = FM1
#OBJECT TYPE = SHELL
#DATA TYPE = PROMAP_40
#SECTION TYPE = PARABOLA
#COMPANY = MEDALLARIO
#REGION = MIDDLE
#NUMBER OF MEASUREMENT = 1
#PRODUCTION NUMBER = 2

#----------------------------------------------------------
#
# PROMAP 40 PSDF SHELL DATA
#
# This corresponds to the file :
# data-root/FM1/PROMAP/023/sf23213a.psd
#
#----------------------------------------------------------
```

```
#AZIMUTH =    000
#ID MANDREL =    023
#MIRROR MODEL = FM1
#OBJECT TYPE = SHELL
#DATA TYPE = PSDF PROMAP_40
#SECTION TYPE = PARABOLA
#COMPANY = MEDALLARIO
#REGION = MIDDLE
#NUMBER OF MEASUREMENT = 1
#PRODUCTION NUMBER = 2

#-

#----------------------------------------------------------
#
# PROMAP 2.5 PSDF SHELL DATA
#
# This corresponds to the file :
# data-root/FM1/PROMAP/023/sp23213a.psd
#
#----------------------------------------------------------

#AZIMUTH =    000
#ID MANDREL =    023
#MIRROR MODEL = FM1
#OBJECT TYPE = SHELL
#DATA TYPE = PSDF PROMAP_25
#SECTION TYPE = PARABOLA
#COMPANY = MEDALLARIO
#REGION = MIDDLE
#NUMBER OF MEASUREMENT = 1
#PRODUCTION NUMBER = 2

#----------------------------------------------------------
#
# UPMC AXIAL MANDREL DATA
#
# This corresponds to the file :
# data-root/MANDREL/UPMC/001/m120.par
#
#----------------------------------------------------------

#AZIMUTH =    120
#ID MANDREL =    001
#OBJECT TYPE = MANDREL
#MIRROR MODEL = MANDREL
#DATA TYPE = UPMC_AXIAL
#SECTION TYPE = PARABOLA
```

```
#-----------------------------------------------------------
#
# UPMC AXIAL SHELL DATA
#
# This corresponds to the file :
# data-root/FM1/UPMC/001/s120.par
#
#-----------------------------------------------------------

#AZIMUTH =    120
#ID MANDREL =    001
#OBJECT TYPE = SHELL
#MIRROR MODEL = FM1
#DATA TYPE = UPMC_AXIAL
#SECTION TYPE = PARABOLA

#-----------------------------------------------------------
#
# UPMC AZIMUTHAL MANDREL DATA
#
# This corresponds to the file :
# data-root/MANDREL/UPMC/001/kreis-6.asc
#
#-----------------------------------------------------------

#AZIMUTH =    6
#ID MANDREL =    001
#MIRROR MODEL = MANDREL
#DATA TYPE = UPMC_AZIMUTHAL

#-----------------------------------------------------------
#
# CMD MANDREL DATA
#
# This corresponds to the file :
# data-root/MANDREL/CMD/024/h24105
#
#-----------------------------------------------------------

#AZIMUTH =    5
#ID MANDREL =    024
#SECTION TYPE = HYPERBOLA
#OBJECT TYPE = MANDREL
#MIRROR MODEL = MANDREL
#DATA TYPE = CMD

#-----------------------------------------------------------
```

```
#
# PROMAP 2.5 MANDREL DATA
#
# This corresponds to the file :
# data-root/MANDREL/PROMAP/024/24h00m0.asc
#
#---------------------------------------------------------

#AZIMUTH =    000
#ID MANDREL =    024
#MIRROR MODEL = MANDREL
#DATA TYPE = PROMAP_25
#COMPANY = ZEISS
#OBJECT TYPE = MANDREL
#SECTION TYPE = HYPERBOLA
#REGION = MIDDLE

#---------------------------------------------------------
#
# PROMAP 20 MANDREL DATA
#
# This corresponds to the file :
# data-root/MANDREL/PROMAP/024/x24p00m0.asc
#
#---------------------------------------------------------

#AZIMUTH =    000
#ID MANDREL =    024
#MIRROR MODEL = MANDREL
#COMPANY = ZEISS
#DATA TYPE = PROMAP_20
#SECTION TYPE = PARABOLA
#REGION = MIDDLE
```