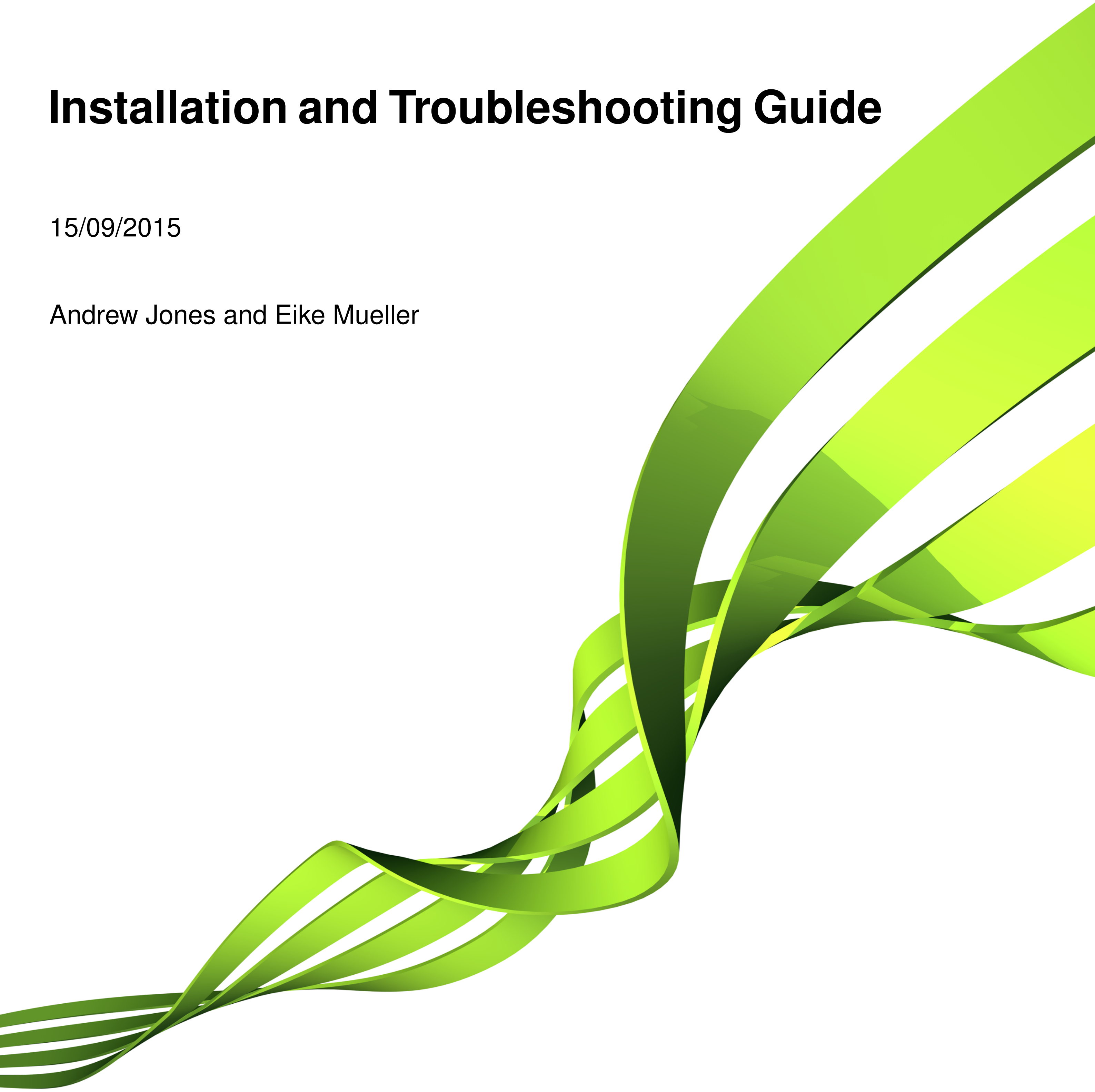
Met Office

Installation and Troubleshooting Guide

15/09/2015

Andrew Jones and Eike Mueller



NAME III Document
MD15/6(version 2)
15/09/15

Installation and Troubleshooting Guide

Andrew R Jones and Eike Mueller

1 Introduction

This document describes the installation procedure for the NAME model, including instructions for testing your NAME installation. It also discusses some of the problems that might be encountered when the model is run on your particular system. The troubleshooting guide suggests solutions for some of the more common issues that have been experienced by external users when setting up NAME for the first time.

2 Supported platforms

The NAME dispersion model is written in the Fortran 95 programming language[1]. The Met Office has built, tested and supports NAME using the Intel Fortran compiler running under the RedHat Linux and Microsoft Windows operating systems. Version 11.0 or later of the Intel compiler is currently recommended as the preferred compiler (there are some known issues when compiling NAME with version 10 of the Intel Fortran compiler). Executable files for both Linux and Windows operating systems are provided as part of the standard build in the NAME distribution. These executables are appropriate for most external users of NAME, and their use is generally recommended. Use of a pre-built executable also simplifies the installation process and avoids the requirement for a Fortran compiler to be available on the installation platform.

When appropriate, a NAME executable can be compiled from its source code. For example, this would be necessary when using a different operating system to the supported operating systems listed above, or when changes have been made to the source code (e.g. by research users of the model). In principle, NAME should be able to run on any platform for which a suitable Fortran compiler is available.

[1] Note that the graphics plotting routines distributed with NAME require access to the data visualisation software package *IDL* (a full licensed version of IDL or an IDL Virtual Machine environment). The testing package also includes some experimental python scripts for plotting two-dimensional NAME field outputs. Python is a free, open source cross-platform programming language. External users may prefer to use their own visualisation software instead for the plotting of NAME outputs.

Any external users intending on compiling the NAME source code can seek further guidance. To date, NAME has also been compiled and run under the following operating systems:

- Sun Solaris O/S
- AIX on IBM HPC
- Cray Linux Environment on Cray HPC
- Apple Macintosh

3 System requirements

System requirements for the installation of the NAME model are minimal in the context of modern computer systems. The full distribution of NAME III requires 750 MB of free space on the hard disk (this includes approximately 500 MB of sample NWP met data files). It is recommended to copy the full distribution, however this contains the NWP met data, multiple builds of the executable applications (release / debug, serial / parallelised, etc.) as well as a significant volume of topography data suitable for use with various versions of the Met Office's Unified Model. If local disk space is an issue then a more selective installation is possible.

System requirements for running NAME are largely determined by the modelling application being considered. NAME can be run in a wide variety of configurations, which influence the load placed on the host computer. An illustration of typical computational demands for various modelling configurations is presented in Table 1. Model runs using NWP met data can require significantly more computing resources than when using simple single-site met data. Guidance on the size of the available NWP data sets is provided in separate documentation. Resources are also determined by factors such as the number of particles in the simulation, the size of the geographical domain and duration of the simulation period being considered, the number of output requests, and the use of more complex modelling options such as chemistry.

Application	*Typical met data*	*Typical RAM usage (GB)*
Short-range emergency response (1 km – 200 km, single source)	Local observations or high resolution NWP data	0.3 - 2.0
Long-range emergency response (100's km – 1000's km, single source)	Global NWP data	0.7 - 4.0
Air quality (multiple species/sources with modelling of atmospheric chemistry)	Global NWP data (European region)	2.0 - 6.0

Table 1: Some typical configurations of the NAME model

Note that the amount of RAM that a single application can address in 32-bit operating systems is limited[2]. In theory the limit is 3 GB, but in practice we have found the limit to be just under 2 GB on both Linux and Windows systems. The recommendation is therefore to limit NAME applications to less than 2 GB on 32-bit operating systems, and to use 64-bit systems whenever applications are likely to exceed this limit.

Model run time is determined by the demands of the simulation, but is also dependent on other factors such as parallelisation options, the CPU speed and other hardware influences. NAME run times can be as short as a few tens of seconds but can extend to several months of computing time. NAME has been run over a broad spectrum of computing systems, from a lowly laptop to a state-of-the-art high-end Linux server or supercomputing platform. As a general rule, a more powerful system will enable larger and faster NAME runs to be performed.

Provided that there is enough RAM available on a system, then it is quite possible to launch multiple NAME runs on the same computer. Our general advice is that the total number of threads allocated for NAME runs (including parallelised NAME runs) should not normally exceed the number of available cores. We have generally found that the optimum number of threads for NAME is influenced by the type of modelling application and also varies from system to system. It can only really be determined for a particular configuration through experimenting with the set up.

4 NAME installation

The installation instructions given here assume that a pre-built executable file is being used. Further advice should be sought from us in instances where the source code is being compiled.

Installation of the NAME model is a simple and easy procedure: **the NAME distribution downloaded from the Met Office FTP server (or supplied on an installation DVD) should be copied to a folder on the host computer**. There is no installation script or wizard, and no requirement for the computer to be restarted after installation. Figure 1 shows the folder structure of the NAME distribution.

Although there is no reason why the installation folder cannot be used to store the local files that you will create, it is generally recommended to set up a separate folders structure to store local files. This should help to ensure that the NAME distribution is not accidentally corrupted. Your local files might include NAME input files (including sources files), met data and output files from your NAME runs.

Use of optional third-party software packages: If NAME is to be used with GRIB or NetCDF format met data files, some additional third-party software also needs to be installed; see the separate guidance on this in the top-level `ReadMe.txt` file.

[2] Of course, there is also a theoretical limit on the memory that can be addressed on 64-bit operating systems, although in practice this is much greater than the total virtual RAM available on current systems.

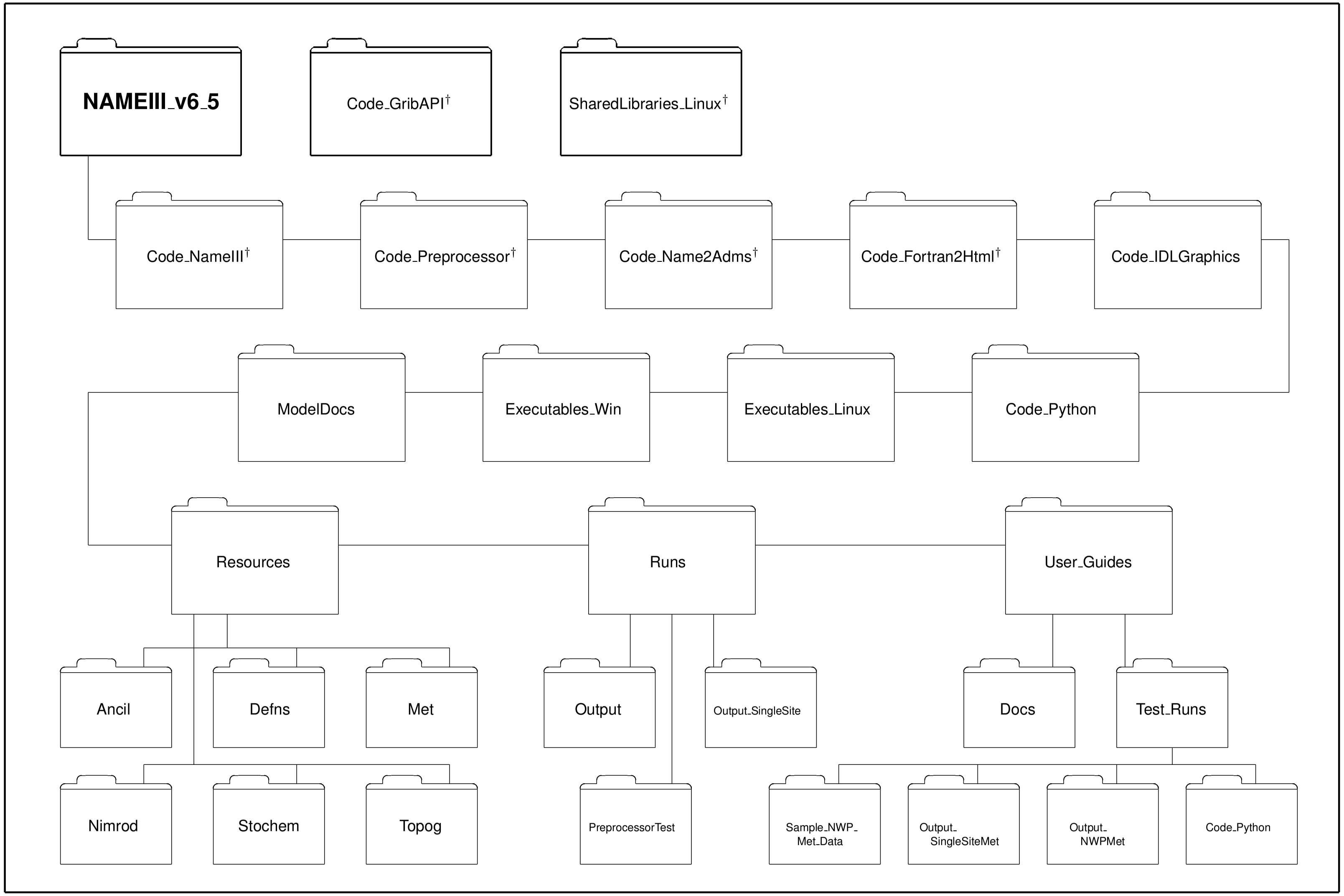


Figure 1: A schematic of the folder structure of the NAME distribution. Folders marked with a dagger[†] are only present when the source code has been supplied with your distribution.

Although it is not essential to set up a local folder structure at the time of installation, it is a good idea to perform this step at that time. One approach might be to follow a similar folder structure as used by the NAME model, or you may prefer to use your own layout. NAME can be very flexible in how it references its input and output files in whatever approach you decide to adopt. Items to consider here include how to arrange your input files for NAME runs, where to write your output files, and how to store your met data files.

5 Testing your NAME installation

Under the **User_Guides** section of the NAME distribution, there is a subfolder $>$ **Test_Runs** which is designed for use in testing your NAME installation. The subfolder contains two example NAME input files and some sample met data for these two scenarios. There is a short-range NAME run that uses single-site met data, and a longer-range NAME run using NWP gridded met data. There are also two sample plotting scripts for generating simple graphics from the typical output files produced by NAME. One is designed to use IDL and the other uses Python, with both scripts illustrating how to plot a two-dimensional spatial field $c(x, y)$. The folder also contains a Linux shell script `RunTests.ksh` and a Windows batch file `RunTests.bat` for performing these test runs automatically on Linux and Windows platforms, respectively, which essentially automates the testing process described below.

The test suite has been designed to use relative pathnames for all file references. It should therefore work correctly in any NAME installation provided that the internal folder structure has not been modified. To test your installation of NAME, run the appropriate test script for your platform or otherwise follow the instructions given here. Note that you might need to change the file permissions on the test scripts to ensure they are 'executable' (e.g. *chmod 755* under Linux).

5.1 Linux platforms

1. Open an interactive shell prompt.

2. The sample NWP met data files stored within the subfolder **User_Guides** $>$ **Test_Runs** $>$ **Sample_NWP_Met_Data** have been gzipped to save disk space and need to be unzipped before use. On a Linux system, use the *gunzip* command.

3. Increase the stack memory size available to the NAME executable using the *ulimit* command, e.g. *ulimit -s unlimited*

4. Change the working directory to the **User_Guides** $>$ **Test_Runs** folder. Then run the NAME executable for the single-site met example using the command

```
../../Executables_Linux/nameiii.exe Example_SingleSiteMet.txt
```

You should see NAME running with various messages written to the terminal (see screenshot in Figure 2). NAME should complete with the message "*Run completed successfully at <clock time>*".

5. NAME should have created some output files in the **User_Guides** > **Test_Runs** > **Output_SingleSiteMet** subfolder. Take a look in this directory and you will find the following set of files:

Fields_grid1_C1_*.txt	Concentration and deposition fields at 3-hourly frequency
Fields_grid2_C1_*.txt	Total integrated concentration field at end of model run
TimeSeries_C1.txt	Concentration time series at specified receptor locations
MetOutput_C1_*.txt	Meteorological time series at specified locations
Info_C1.txt	Other information on the state of the model run

All NAME output files are in plain text ASCII format, which can be opened for viewing in any file editor or converted to graphical products using your favourite plotting package.

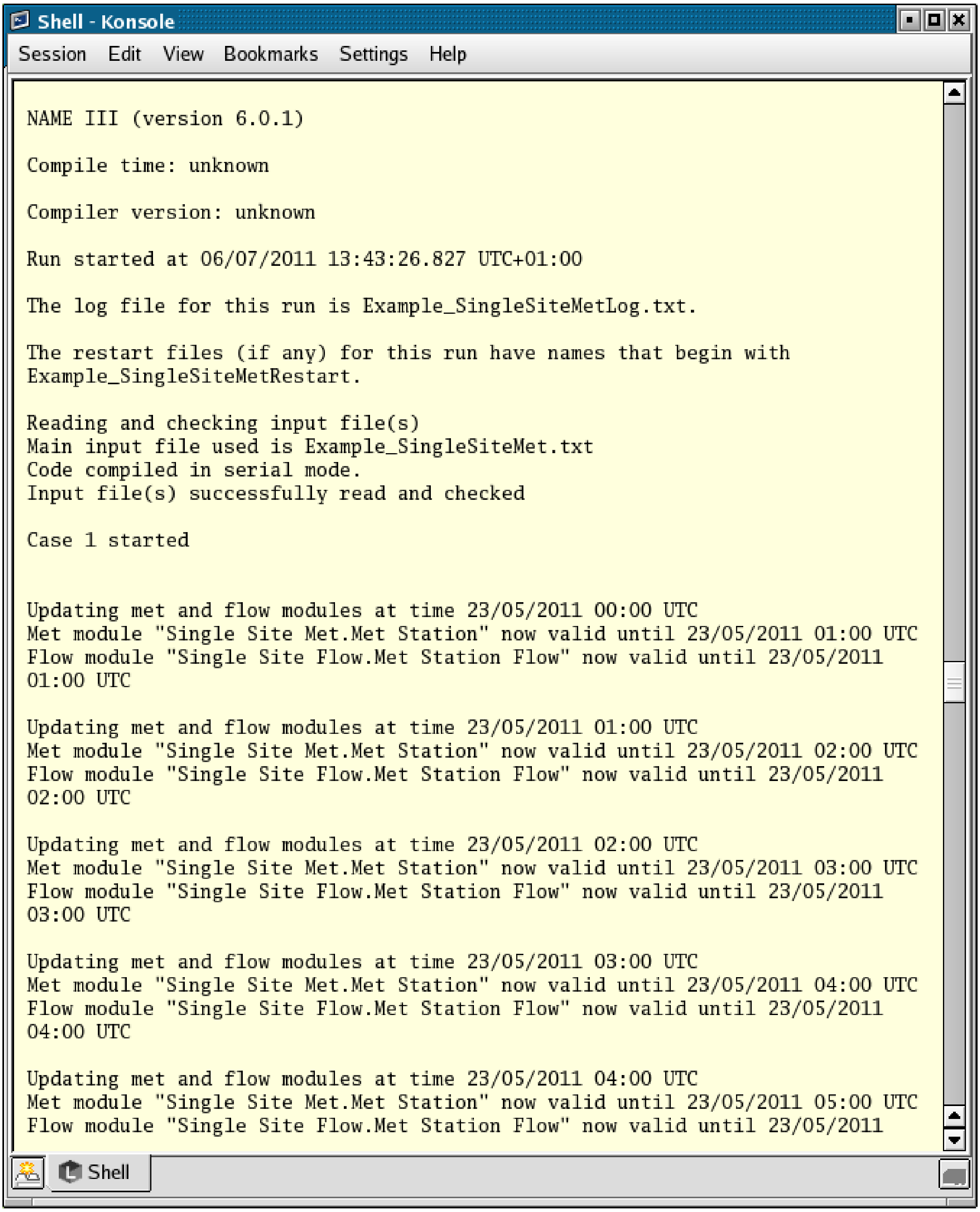
6. (Optional) If IDL or Python is installed on your system, graphics can be plotted using the supplied scripts `genIDLplots.ksh` or `genPythonPlots.ksh`. Both scripts are called with a single argument, viz. the absolute path of the directory containing the NAME output files

```
./genIDLplots.ksh /home/.../User_Guides/Test_Runs/Output_SingleSiteMet

./genPythonPlots.ksh /home/.../User_Guides/Test_Runs/Output_SingleSiteMet
```

The IDL code requires version 7.1 (or later) of IDL. The Python code requires version 2.5 (or later) of Python 2 (but does not work with Python 3). See the next section for instructions on downloading and installing Python 2.x if this is not already installed on your system. The IDL script generates its output graphics in the same folder as the NAME output files. By default, the same is also true for the Python script (although this behaviour can be easily modified to use a separate subfolder if desired). Plot and then view the graphics.

The use of single-site met data should be restricted to the modelling of relatively short-range dispersion problems where horizontal variations in the large-scale meteorology can be reasonably ignored. In this particular example, NAME predicts the dispersion out to approximately 80 km from the source (which is arguably too far really for a homogeneous meteorology assumption, but it does illustrate the concept of running with single-site meteorology). When using NWP met data,



```
NAME III (version 6.0.1)

Compile time: unknown

Compiler version: unknown

Run started at 06/07/2011 13:43:26.827 UTC+01:00

The log file for this run is Example_SingleSiteMetLog.txt.

The restart files (if any) for this run have names that begin with
Example_SingleSiteMetRestart.

Reading and checking input file(s)
Main input file used is Example_SingleSiteMet.txt
Code compiled in serial mode.
Input file(s) successfully read and checked

Case 1 started


Updating met and flow modules at time 23/05/2011 00:00 UTC
Met module "Single Site Met.Met Station" now valid until 23/05/2011 01:00 UTC
Flow module "Single Site Flow.Met Station Flow" now valid until 23/05/2011
01:00 UTC

Updating met and flow modules at time 23/05/2011 01:00 UTC
Met module "Single Site Met.Met Station" now valid until 23/05/2011 02:00 UTC
Flow module "Single Site Flow.Met Station Flow" now valid until 23/05/2011
02:00 UTC

Updating met and flow modules at time 23/05/2011 02:00 UTC
Met module "Single Site Met.Met Station" now valid until 23/05/2011 03:00 UTC
Flow module "Single Site Flow.Met Station Flow" now valid until 23/05/2011
03:00 UTC

Updating met and flow modules at time 23/05/2011 03:00 UTC
Met module "Single Site Met.Met Station" now valid until 23/05/2011 04:00 UTC
Flow module "Single Site Flow.Met Station Flow" now valid until 23/05/2011
04:00 UTC

Updating met and flow modules at time 23/05/2011 04:00 UTC
Met module "Single Site Met.Met Station" now valid until 23/05/2011 05:00 UTC
Flow module "Single Site Flow.Met Station Flow" now valid until 23/05/2011
```

Figure 2: Screen shot illustrating the messages output at the start of a NAME run using single-site met data.

the meteorological fields are spatially varying and longer-range problems can be represented, as in the next example where we model the transport and dispersion across a northern European domain.

7. Next, run the NAME executable for the NWP met example using the command

   ```
   ../../Executables_Linux/nameiii.exe Example_NWPMet.txt
   ```

   You should again see NAME running with various messages written to the terminal as before (see screenshot in Figure 3). NAME should again complete with the message "*Run completed successfully at <clock time>*".

8. For the NAME run using NWP met data, NAME should generate a similar set of output files to those in the single-site example above, but this time the files are written to the **User_Guides** > **Test_Runs** > **Output_NWPMet** subfolder. Take a look in this directory and you will find the files.

9. (Optional) Plot and view the graphics as before.

5.2 Windows platforms

1. Open a DOS command prompt.

2. The sample NWP met data files stored within the subfolder **User_Guides** > **Test_Runs** > **Sample_NWP_Met_Data** have been gzipped to save disk space and need to be unzipped before use. On a Windows system, use the *WinZip* utility.

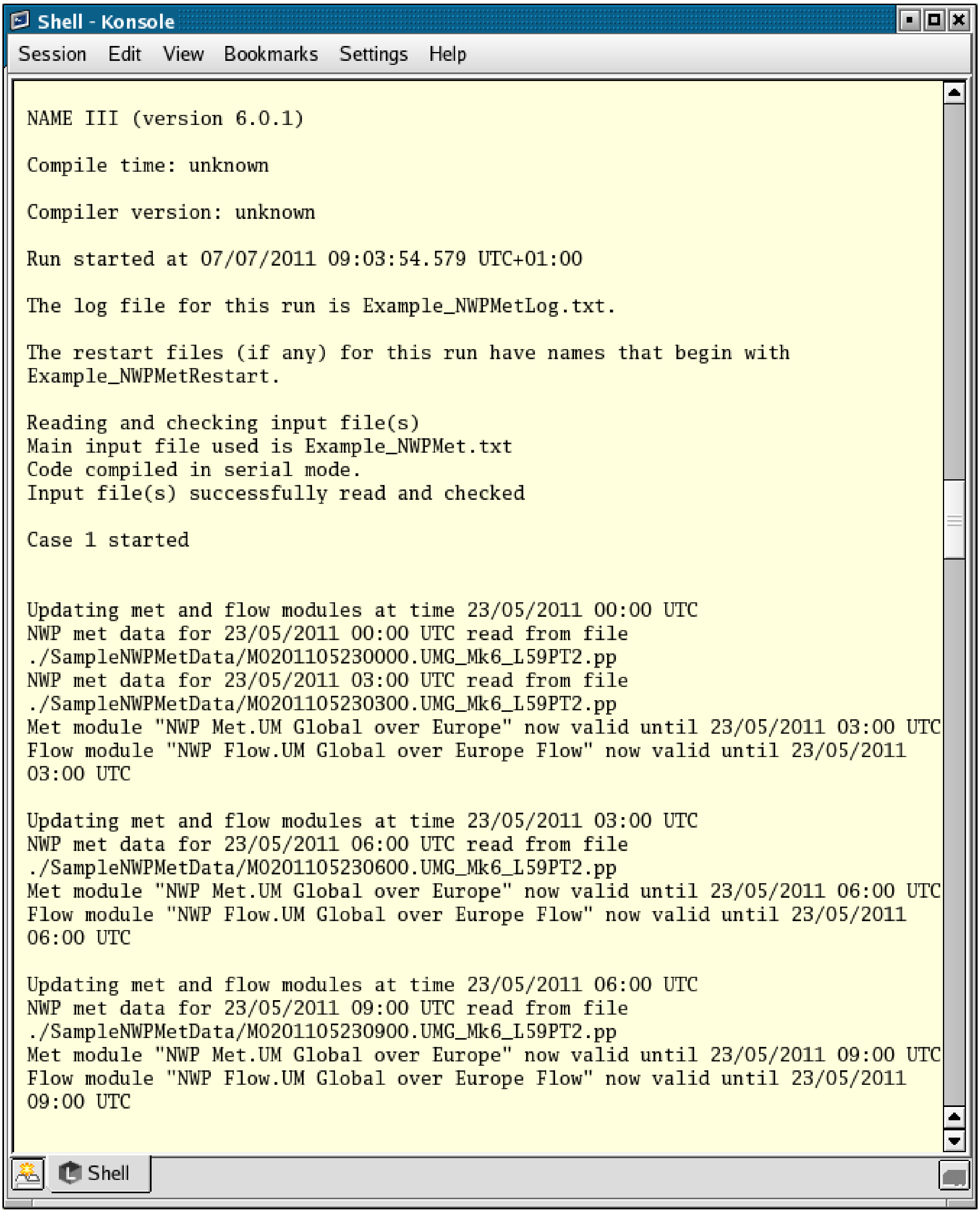
3. Change the working directory to the **User_Guides** > **Test_Runs** folder. Then run the NAME executable for the single-site met example using the command

   ```
   ..\..\Executables_Win\nameiii.exe Example_SingleSiteMet.txt
   ```

   You should see NAME running with various messages written to the terminal (similar to the screenshot shown in Figure 2). NAME should complete with the message "*Run completed successfully at <clock time>*".

4. NAME should have written some output files in the **User_Guides** > **Test_Runs** > **Output_SingleSiteMet** subfolder. Take a look in this directory and you should find the following set of files:

Fields_grid1_C1_*.txt	Concentration and deposition fields at 3-hourly frequency
Fields_grid2_C1_*.txt	Total integrated concentration field at end of model run



```
NAME III (version 6.0.1)

Compile time: unknown

Compiler version: unknown

Run started at 07/07/2011 09:03:54.579 UTC+01:00

The log file for this run is Example_NWPMetLog.txt.

The restart files (if any) for this run have names that begin with
Example_NWPMetRestart.

Reading and checking input file(s)
Main input file used is Example_NWPMet.txt
Code compiled in serial mode.
Input file(s) successfully read and checked

Case 1 started


Updating met and flow modules at time 23/05/2011 00:00 UTC
NWP met data for 23/05/2011 00:00 UTC read from file
./SampleNWPMetData/MO201105230000.UMG_Mk6_L59PT2.pp
NWP met data for 23/05/2011 03:00 UTC read from file
./SampleNWPMetData/MO201105230300.UMG_Mk6_L59PT2.pp
Met module "NWP Met.UM Global over Europe" now valid until 23/05/2011 03:00 UTC
Flow module "NWP Flow.UM Global over Europe Flow" now valid until 23/05/2011
03:00 UTC

Updating met and flow modules at time 23/05/2011 03:00 UTC
NWP met data for 23/05/2011 06:00 UTC read from file
./SampleNWPMetData/MO201105230600.UMG_Mk6_L59PT2.pp
Met module "NWP Met.UM Global over Europe" now valid until 23/05/2011 06:00 UTC
Flow module "NWP Flow.UM Global over Europe Flow" now valid until 23/05/2011
06:00 UTC

Updating met and flow modules at time 23/05/2011 06:00 UTC
NWP met data for 23/05/2011 09:00 UTC read from file
./SampleNWPMetData/MO201105230900.UMG_Mk6_L59PT2.pp
Met module "NWP Met.UM Global over Europe" now valid until 23/05/2011 09:00 UTC
Flow module "NWP Flow.UM Global over Europe Flow" now valid until 23/05/2011
09:00 UTC
```

Figure 3: Screen shot illustrating the messages output at the start of a NAME run using NWP met data.

TimeSeries_C1.txt Concentration time series at specified receptor locations

MetOutput_C1_*.txt Meteorological time series at specified locations

Info_C1.txt Other information on the state of the model run

All NAME output files are in plain text ASCII format, which can be opened for viewing in any file editor or converted to graphical products using your favourite plotting package.

5. (Optional) Plot graphics.

   The use of single-site met data should be restricted to the modelling of relatively short-range dispersion problems where horizontal variations in the large-scale meteorology can be reasonably ignored. In this particular example, NAME predicts the dispersion out to approximately 80 km from the source (which is arguably too far really for a homogeneous meteorology assumption, but it does illustrate the concept of running with single-site meteorology). When using NWP met data, the meteorological fields are spatially varying and longer-range problems can be represented, as in the next example where we model the transport and dispersion across a northern European domain.

6. Next, run the NAME executable for the NWP met example using the command

   ```
   ..\..\Executables_Win\nameiii.exe Example_NWPMet.txt
   ```

   You should again see NAME running with various messages written to the terminal as before (similar to the screenshot shown in Figure 3). NAME should again complete with the message "*Run completed successfully at <clock time>*".

7. For the NAME run using NWP met data, NAME should generate a similar set of output files to those in the single-site example above, but this time the files are written to the **User_Guides** > **Test_Runs** > **Output_NWPMet** subfolder. Take a look in this directory and you will find these files.

8. (Optional) Plot and view the graphics as before.

6 Plotting with python

Two-dimensional fields $c(x, y)$ contained in the NAME output files can be plotted using the python script `PlotField2d.py` (which uses the *Field2d* class in `Field2d.py`). This script is experimental and only provides minimal functionality sufficient for plotting the examples given here. We cannot guarantee that it will work under general circumstances. The user should feel free to adapt it for their particular needs or to develop its functionality further.

6.1 Requirements

The following software is needed to run the python scripts for plotting NAME output:

- **python** (version 2.5 or later – but NOT 3.x),

  which can be downloaded from http://www.python.org
- **numpy** package,

  which can be downloaded from http://numpy.scipy.org/
- **matplotlib** package,

  which can be downloaded from http://matplotlib.sourceforge.net/
- **matplotlib** basemap toolkit,

  which can be downloaded from http://matplotlib.sourceforge.net/basemap/doc/html/

6.2 Running the python script

The python script `PlotField2d.py` plots a specified field stored within a given set of NAME output files (as specified by their 'grid'). The script is invoked as follows:

```
python PlotField2d.py --datadir=<datadir>  --plotdir=<plotdir>  --grid=<grid>
                      --fieldname=<fieldname>  --label=<label>
```

where the following command line options need to be specified,

`<datadir>` *The directory containing the NAME output files*

`<plotdir>` *The directory into which the graphics files are to be written*

`<grid>` *The name of the grid used by the NAME output files (e.g. to plot data contained in the files* Fields_grid1_*.txt *this should be set to* `--grid="grid1"`*)*

`<fieldname>` *The name of the field to be plotted, as specified in the column header (e.g.* `--fieldname="Total Deposition"`*)*

`<label>` *The prefix for the output graphics files (e.g. if set to* `--label="TotalDeposition"`*, the output files will be of the form* 'TotalDeposition_TXXXX.png' *where* XXXX $= 0001, 0002$*, etc.)*

7 Troubleshooting problems with NAME

The NAME model is run by launching its executable file. This can be done in a variety of different ways: by double clicking its application icon[3] (Windows only), by invoking at the command line, or by running from a script. Most users find launching NAME from within a script to be the most useful approach. This is also the way we usually run NAME within the Met Office.

A number of common problems have occasionally been encountered by external users when they try to use NAME for the first time. These are described in the troubleshooting guide on the following pages, which on encountering any problems should be consulted in the first instance. Please feedback any further items which you think would be useful to add to this list.

[3] For non-Intel compilers and older versions of the Intel compiler, there may be a potential issue with endian conversion (see page 14). This can be resolved by running from a batch file to set the endian conversion prior to launching NAME.

Problem	Possible solution
On Linux systems, NAME fails at launch with a memory fault	The most likely cause of this problem is an insufficient size for stack memory. This can be corrected by using the **ulimit** command to increase the stack size, viz. • **ulimit -s unlimited**, or • **ulimit -s** N for an explicit size N. The stack size should be specified prior to invoking NAME, e.g. within a script, or alternatively it may be declared within a user's `.profile` or `.kshrc file`. Similarly, when running with multiple threads under OpenMP, it is recommended to set the environment variable `OMP_STACKSIZE` `export OMP_STACKSIZE='4m'` This defines the stack size that is allocated for any additional threads created during the NAME run. The default, if not set explicitly, is 4 MB ('4m') which is usually sufficient for a parallelised NAME run but it may be necessary to set a larger value than this when global parameters have been modified by the user.
Under Windows, NAME fails at launch with a system error message stating that `libguide40.dll` could not be found (or similar messages for other libraries that might be missing)	Missing .dll files can be an issue when running with Open MP on a Windows machine that does not have the Intel Fortran compiler installed (because some of the required Intel library files are not installed as standard). The problem can be resolved by downloading and installing the redistributable package that includes the required runtime .dll from the Intel website at http://software.intel.com/en-us/articles/redistributable-libraries-of-the-intel-c-and-fortran-compiler-for-windows/
NAME reports a problem with the input file even though it appears to be correct	Check for use of TAB characters in the NAME input files. TAB characters should **not** be used in any NAME input file as they can have unpredictable consequences. Hint: TAB characters can be highlighted in the editor `NEdit` by using the `Preferences > Apply Backlighting` option. Problems can also arise when copying files between Windows and Linux platforms due to a format difference in text files (Windows files contain an extra 'line-feed' at the end of each line). Some transfer mechanisms automatically convert the format (e.g. ftp) but others may not. Any problems can be resolved by explicitly converting the file format using the pair of Linux utilities **dos2unix** and **unix2dos**. The file format issue also affects single-site met files (e.g. if they were originally created from an Excel spreadsheet, say).

NAME successfully reads and checks the input file(s) but fails when attempting to read an NWP topography file	This is possibly an endian issue where the native endian on your system is different to the endian used by the NWP met and topography data (this causes an error when attempting to read such files in unformatted I/O statements). Big endian format is used for NWP meteorological data from the Met Office's Unified Model. Endian conversion is performed automatically for some compilers (e.g. more recent versions of the Intel compiler) but is not supported for all compilers. Endian conversion can be invoked explicitly by setting the environment variable `F_UFMTENDIAN`, e.g. • `F_UFMTENDIAN = "big"` to set the default behaviour of the system to apply little-endian to big-endian conversion on all logical units, or • `F_UFMTENDIAN = "big;little:N"` to apply conversion on all logical units with the exception that endian conversion is not performed on logical unit N. In practice, the user should set this environment variable with **F_UFMTENDIAN = "big;little:100"** to ensure the correct behaviour appropriate to the NAME model. The reason for reserving unit number 100 on which the conversion is not done is associated with *restartable* runs. The information contained within NAME restart files includes some derived types. However, little-endian to big-endian conversion is not supported for derived types in unformatted I/O. The restart file (reserved unit 100) must therefore be written in the native endian. a) On linux systems, type **export F_UFMTENDIAN="big;little:100"** to set the environment variable prior to running the NAME executable. b) On Windows, type **set F_UFMTENDIAN=big;little:100** at the DOS prompt or in a batch file prior to calling the NAME executable. The above command appears to be sensitive to case on different versions of the Windows operating system, and users may need to experiment with lower case and upper case variants of this text.

NAME is unable to read gzipped NWP met data files	The current version of NAME is not able to read gzipped met files directly (although the intention is to explore the possibility of including this as an option in the future). Instead, all NWP met data files need to be unzipped before they are read by NAME. When using a met restore script with NAME, the met restore process needs to ensure that each met file is restored in an unzipped form (typically, this might involve copying a gzipped file from a met archive to a local met folder and then unzipping that local duplicate). When reading NWP met data directly from a met folder (i.e. not using a met restore script), the user should ensure that all the required met files are available unzipped prior to running NAME. a) On linux systems, file zipping/unzipping is done using the general Linux file utility **gunzip**. You should ensure that the gunzip utility is available on your system (although this should be part of any standard Linux build). b) On Windows, file zipping/unzipping can be done using the WinZip utility. Note that NWP met files from the Met Office's Unified Model have high compression (the -9 option in gzip) from 2009 onwards and used standard compression (-5 option) before that time.
A met restore script cannot unzip the gzipped data files	Ensure that a suitable unzip utility is available on your system. a) On linux systems, use the general file utility **gunzip** (which should be part of any standard Linux build). b) On Windows, use the WinZip utility.

For the GRIB-API version of NAME (available for Linux only):
NAME is unable to find the GRIB-API shared object libraries at run time and reports the error message *"error while loading shared libraries: libgrib_api_f90.so: cannot open shared object file: No such file or directory"*.
	The location of the GRIB-API .so libraries needs to be added to the environment variable `LD_LIBRARY_PATH` before running the GRIB-API version of NAME, viz.
	`export LD_LIBRARY_PATH =`
	`$LD_LIBRARY_PATH:[GRIB_API_LIB]`
	Similarly, when reading NetCDF met data, the NetCDF API libraries need to be included in the `LD_LIBRARY_PATH` search path.

For the GRIB-API version of NAME (available for Linux only): The GRIB-API does not correctly interpret the contents of GRIB format met files.	The following environment variables need to be set to correctly configure the GRIB-API at run time: • `GRIB_DEFINITION_PATH = ${SHAREDLIB_DIR}/share/definitions` • `GRIB_SAMPLES_PATH = ${SHAREDLIB_DIR}/share/samples` • `GRIB_API_INCLUDE = ${SHAREDLIB_DIR}/include` • `GRIB_API_LIB = ${SHAREDLIB_DIR}/lib` where `${SHAREDLIB_DIR}` is the top-level directory of the GRIB-API installation.

Met Office
FitzRoy Road, Exeter
Devon, EX1 3PB
UK

Tel: 0370 900 0100
Fax: 0370 900 5050
enquiries@metoffice.gov.uk
www.metoffice.gov.uk

Produced with the Met Office LaTeX template v2.0.0