

CU-QMW-MA-0007

Date: 21 Mar 2007

Issue: 2

Rev. : 3.0

Page: i

Installation Notes for QSAS

A.J. Allen

The Blackett Laboratory
Imperial College London

London SW7 2AZ

e-mail

csc-support-dl@imperial.ac.uk

Document Status Sheet			
1. Document Title: QSAS Installation			
2. Document Reference Number: CU-QMW-MA-0007			
3. Issue	4. Revision	5. Date	6. Reason for Change
1	0	13 March 1996	QSAS R1.0, Static build only
1	1	18 June 1997	QSAS R1.2, improved troubleshooting
1	1.1	15 July 1997	now at cdf 2.6
1	1.2	15 September 1997	Add QRTN plug-in and idl source.
1	1.3	17 December 1998	Supports idl4 and 5
1	1.4	22 January 1999	Extra trouble shooting info
1	1.5	1 December 1999	For QSAS R1.4. Source release
1	1.6	16 November 2000	QSAS R1.5 under gnu and pgplot
1	1.8	14 March 2001	QSAS R1.54 Added Linux support
1	1.11	26 October 2001	QSAS R1.61 new Data selector
1	1.12	8 May 2002	QSAS R1.64
1	1.13	6 Sep 2002	QSAS R1.66 - fixes bug, Boxcar for Scalar Seq
2	0.1	19 Dec 2003	QSAS R2.0.0 - Major release
2	0.2	23 March 2004	QSAS R2.0.02 Simpler install, faster plot, new qtpl parser.
2	1.0	9 Dec 2005	QSAS R2.1.10 Calculator and enhanced plotting.
2	1.1	23 Jan 2006	Includes note on building on 64 bit platforms
2	2.0	12 May 2006	Major release 2.2.00, Mac port and use of configure.
2	2.1	22 January 2007	Minor release 2.2.01, Mac intel port and plugin lib fixes
2	2.2	21 March 2007	Changes to default autoconf installation
2	3.0	9 December 2008	New release ported to Windows, plplot and new qpui

Reference Documents

1. “QSAS Help Pages”, CU-QMW-MA-0014
2. “Reference Document for CSDS CDF Implementation, DS-QMW-TN-0003”
3. “Writing QSAS Plug-ins, CU-QMW-TN-0008”
4. “Cluster Exchange Format, DS-QMW-TN-0010”

Contents

1	Introduction	1
2	Host System Requirements	1
3	Fetching QSAS	1
4	Installing QSAS	2
4.1	Installing under Solaris or Linux	2
4.2	Incremental Updates for configure Installation	3
5	Installation using old build script method	4
5.0.1	User Configuration	5
6	Installing on Mac OSX	6
7	Notes on QSAS Source Build using the build script	7
7.1	Compilers	7
7.2	Pgplot	7
7.3	CDF library	8
7.4	Lapack and Blas	9
7.5	Linux	9
7.6	QT	9
7.7	Compiling under Mac OSX	9
7.8	Compiling local Plug-ins	9
8	Troubleshooting	11
8.1	Launching qsas	11
8.1.1	Execution error	11
8.1.2	Execution error	11
8.1.3	qsas file read/write errors	11
8.1.4	No QSAS GUI window	12
8.2	Data Access	12
8.2.1	QDS cannot get data	12
8.2.2	QDS cannot get data	12
8.2.3	qsas crashes on data access	12
8.3	Plug-ins	13

8.3.1	Cannot run plugin written for QSAS 1.6	13
8.3.2	Cannot find Plug-in entry point	13
8.3.3	Plugin fails to parse or run.	13
8.3.4	Locally written plug-in requires debugging	13
8.4	Reporting problems	14

1 Introduction

This document provides information on how to fetch and install the UK Cluster Science Centre data analysis software package, QSAS (QM Science Analysis System).

QSAS is provided free under the gpl public licence
see <http://www.space-plasma.qmul.ac.uk/QSAS/QSASlicence.html>

2 Host System Requirements

QSAS has been tested on Sun Solaris under Common Desktop Environment(CDE), Linux under KDE, Mac OSX (10.4 on G4, G5 and Intel processors) and Windows (XP and Vista).

The third party software QT (version 4.4, but QT3 is still supported) is required, and is also available under the gnu public licence.

3 Fetching QSAS

The full QSAS distribution is available via the following address:

http://www.space-plasma.qmul.ac.uk/QSAS/qsas_welcome.html

Documentation may be browsed or fetched independently of the QSAS software.

Different downloads are available for Linux/Solaris, Mac OSX and Windows. The online help pages are included with the distribution (**QSAS home/doc/html/index.html**) and on the QM web site (<http://www.space-plasma.qmul.ac.uk/QSAS/QSAShelp>). The help pages may be browsed using a stand-alone browser or by selecting the help menu within QSAS.

4 Installing QSAS

Instructions for installing QSAS and QT4 are provided below for each platform in turn.

4.1 Installing under Solaris or Linux

It is necessary to ensure that QT is installed before QSAs as autoconf for the QSAS installation will look for it. QT is installed as part of KDE in most linux systems, but will probably need installing for on Solaris.

If QT is not installed, we recommend using QT4 from...

<http://trolltech.com/downloads/opensource/appdev>

To install QSAS, you need to perform the following steps....

To uncompress 'qsas-2.3dist.tar.gz' use the 'gzip' command...

```
gzip -d qsas-2.3dist.tar.gz
```

The resulting file will be named 'qsas-2.3dist.tar'.

Then use the 'tar' command to extract the QSAS source files. 'tar' will create a directory named 'qsas-2.3dist' containing several subdirectories. The source files for the distribution will be placed into these directories.

The syntax for the 'tar' command is different on some flavours of UNIX. For most UNIX systems use...

```
tar -xvf qsas-2.3dist.tar
```

The following steps should then automatically build and install QSAS.

1. Set your current directory (cd) to the top level QSAS directory (the directory named 'qsas-2.3dist' created by 'tar').

2. In order to configure QSAS to install into a directory /home/mydir/QSAS_2.3 it should be necessary only to type the following command...

```
./configure --prefix=/home/mydir/QSAS_2.3
```

This will search all the usual locations (/usr/lib, /usr/local/lib, /lib etc) for the QT and system libraries. This may take a few minutes.

If unable to locate a usable version of QT it will stop with an error message (see step 6 below).

3. If no errors are reported compile and build the QSAS system by running the command

```
make
```

Please be patient here, since a base configuration takes between 15 and 40 minutes to compile depending on your hardware.

4. Install QSAS in the target directory specified in the --PREFIX argument (see --prefix option in step 2) by running...

```
make install-strip
```

NB. If you wish to retain the debugging symbols to use QSAS with a debugger use the following command instead...

```
make install
```

QSAS is now installed in the target directory specified.

To save space after installation you can type

```
make clean
```

5. To run QSAS, execute the script in the bin sub-directory of the target directory, e.g. ...

```
/home/mydir/QSAS_2.3/bin/QSAS
```

This script may be copied into each user's own space or added to their path environment. Individual copies of the QSAS script may be edited to change default locations for placing e.g. plot files or savesets.

6. To specify a specific local version of QT or CDF, or if your local installation of QT is not in one of the 'usual places' there are two ways to tell configure where to look.

In the simplest case you can tell configure to use a specific environment variable, such as the location of QT by adding the environment and its value to the configure command, e.g.

```
./configure --prefix=/home/mydir/QSAS_2.3 QT_DIR=/usr/lib/qt4
```

then configure will look in the directories /usr/lib/qt4 and /usr/lib/qt4/lib for the QT libraries.

QSAS can be forced to use a local installation of CDF by similarly setting the environments CDF_DIR.

The command configure accepts a variety of variables and command line options. Entering...

```
./configure --help
```

will give an overview of these options.

Alternatively, to set several environments at once, modify the shell script file named 'qsas-2.3dist/share/config.site' and specify the locations of the QT or CDF sub-directories for your distribution. These will be pre-pended to the search path used by configure.

It is then necessary to use the CONFIG_SITE environment variable to point to the directory path for the config.site file containing your settings

```
./configure --prefix=/home/mydir/QSAS_2.3 CONFIG_SITE=./share/config.site
```

Once you have run configure successfully go back to step 3 and continue compiling and installing QSAS.

4.2 Installing on Windows

QSAS for Windows is delivered as a stand-alone, compiled package. Unzipping the content of the Windows QSAS archive creates a "QSAS" folder, with several sub-folders. Edit "QSAS.bat" in the "bin" sub-folder with any text editor (e.g. Notepad) and adapt the first line of this file.

If you unzipped QSAS in C:\temp, you should have a C:\temp\QSAS folder, and the first line should be:

```
set QSAS_HOME=C:\temp\QSAS
```

You do not have to change anything if you unzipped QSAS in C:\Program Files.

To start QSAS, double-click "QSAS.bat". It should open, along with a DOS console you can use to check the QSAS outputs.

To uninstall QSAS, remove the QSAS folder. Your settings (profiles) are stored in C:\Documents and Settings\user name\qsas.ini.

You can also install QSAS by compiling it on Windows. For this, download the QSAS source build package (see ??), and follow the instructions in doc/build.windows.txt.

4.3 Installing on Mac OSX

QSAS 2.3 on Mac OSX requires that QT4 is installed. This is available as a disk image from

<ftp://ftp.trolltech.com/qt/source/qt-mac-opensource-4.4.3.dmg>

and this installs a universal binary for both PowerPC or Intel Macs.

The QSAS disk image contains an installer that will place the directory `QSAS_2.3` in the `/Applications` directory and installs universal binaries for both PowerPC and Intel Macs. It should not be moved anywhere else as paths are absolute and not relative. Double clicking the QSAS application will launch QSAS.

It is possible to drag the QSAS application to the Dock or the Favourites pane of the finder, from where it may be launched with a single click.

This executable is in fact the QSAS script, and it will configure and run the binary executable within the QSAS distribution. The CDF libraries are provided with QSAS.

The directory `/Applications/QSAS_2.3` contains a full QSAS Home directory with source, and `/Applications/QSAS_2.3/bin/buildPlugin` is a script to assist in building local plugins.

4.4 Installation from Linux rpm

A limited number of platform specific rpm distributions are available, and may be constructed from the autoconf distribution. We would be happy to place any user created rpm files on our server.

4.5 Notes on QSAS Source Build using the build script under Linux or Mac OSX

This method should only be used in the binary or autoconf installations failed.

It is possible to build QSAS from source using a script, 'build', that sets environment variables and recursively runs make on the source tree. It requires some (free) third party libraries.

It is recommended that the QSAS distribution is kept in a dedicated directory with group read/write access. QSAS will unpack into the current working directory at the time of extraction from the tar file. It will create a top level directory called **QSAS_2.2**. This is the **QSAS_home** directory that must be set in the QSAS script. Subdirectories `src`, `lib`, `doc` and `bin` *etc.* will be generated automatically.

Place the compressed QSAS distribution tar file in the directory that is to be the parent directory of **QSAS_2.2**.

The QSAS distribution is uncompressed using

```
gunzip QSAS_2.2_binary.tar.gz
```

and extracted from the tar file using

```
tar -xpvf QSAS_2.2_binary.tar
```

The tar option `p` is included to ensure that read and write permissions are set according to those on the tar file. If this option is not specified the extracted files will have owner and group access permissions determined by the `umask` of the user performing installation.

The QSAS home directory may be renamed at this point. It will unpack as **QSAS_2.2**

To compile QSAS using the build script change directory to **QSAS_2.2/bin**. Edit the script **QSAS_2.2/bin/build** to set the **QS_HOME** environment to the local directory **QSAS_2.2** of this QSAS installation. Also set paths for the libraries on your system in the build script. At the system prompt type **build qsas**. This will hierarchically compile the source distribution including the supplied plug-ins, and place the executables in **QSAS_2.2/lib**. This will overwrite the supplied executables. This process should take about 30 minutes. When completed type **build clean** to remove object code no longer needed as this can be large.

To make a smaller qsas executable by stripping debug information, type **build strip**. Omit this step if the executable is to be run with a debugger.

It is also possible to do a partial rebuild, **build version** will recompile only components that have changed or failed at a previous make, and will install executables as needed.

Ensure after installation that user write permission is set for **QSAS_2.2/tmp** and also that execute permission is set for **QSAS_2.2/lib/qsas** and **QSAS_2.2/bin/***.

The user who unpacks the tar file will become the owner of the **QSAS** files, but read access is provided for all users.

The file **QSAS** in directory **QSAS_2.2/bin** should now be edited to set the following environment variables:-

QSAS_HOME This environment variable is the full path to the **QSAS_2.2** directory at installation.

LD_LIBRARY_PATH (DYLD_LIBRARY_PATH on mac) This is set locally for use within QSAS to avoid conflicts with other libraries set in the user's own **LD_LIBRARY_PATH**. It temporarily overrides the users own settings, but does not affect any other applications run by the user. If the third party libraries shipped with QSAS are to be used, then the path **QSAS_2.2/lib/syslib** should be pre-pended to the **LD_LIBRARY_PATH**.

CDF_LIB and CDF_BIN These must be set for a source build.

Other installation wide settings may be specified here, and QSAS look and feel can be modified through these environments variables.

QSAS ships with a set of sample Cluster Summary Parameter data to allow users to try out the Data Selector interface. To configure this for use, launch QSAS by running the script **QSAS** and go to the **Data Files:Data Selection** menu item. In the resulting window select **Database:Create Database Entries**.

Under Solaris use the file navigation window produced to locate **SampleClusterData** in **QSAS_2.2** then select the directory **SPdata** and click **OK**. Under Linux use the file navigation window produced to locate the directory **SampleClusterData/SPdata** in **QSAS_2.2** (the field **Look in:** will end in **/SPdata**) and click **Open**.

In the **Data Name Root** window type **SP** and click **OK**. In the **Save As Database** window type **SP.dcm** in the **File name** field and click **Save**. This will map the sample

SP holdings, and when complete will ask you if you want to open the SP.dcm file. Click **Open** to see the sample SP holdings - a few days of data). Only the last day will be selected in the time editor, so navigate backward by changing the start time to see the rest of the holding. For subsequent QSAS sessions it will only be necessary to select **Database:Open** menu item in the **Data Selection** window and select the SP.dcm file.

Note that any directory (or directories) holding cdf data files can be mapped for the Data Selector in the same way, and updated when new data are added (or removed), but the user must have write access to the location chosen for the .dcm map files. See **User Configuration** below, and the QSAS Help Pages.

4.5.1 CDF library

Installation of the CDF library version 2.6, or later, is required for a source build (available from http://nssdc.gsfc.nasa.gov/cdf/cdf_home.html/). QSAS is normally built against cdf version 3.1 or version 2.7. CDF version 3.0 should not be used as it is unable to write backwardly compatible files. QSAS sets the environment variable CDF_FILEBACKWARD to true to ensure it creates files that can be read by cdf versions 2.6 and 2.7.

Download and uncompress the CDF distribution and extract the tar archive in a temporary directory.

This directory will be used to compile the distribution, but a separate installation directory will be created during to installation process.

Detailed installation instructions are provided with the CDF distribution, but for most installations it will be necessary only to do one of the following in the local cdf top level directory.

Under Solaris, type

```
make OS=solaris ENV=gnu SHARED=yes all
```

Under linux, type

```
make OS=linux ENV=gnu SHARED=yes all
```

This will compile the distribution for your platform. Then type

```
make test
```

To install the library locally, make a target directory, e.g. /usr/lib/cdf then type

```
make INSTALLDIR=/usr/lib/cdf install
```

to install the cdf libraries in a folder cdf under /usr/lib. The choice of local path is up to the user, and must be matched by that set in the **build** script and **QSAS_settings** file when building and running QSAS. All QSAS users will need read permission to this distribution.

5 User Configuration

To simplify user configuration and to avoid the setting of paths and environments that may interfere with local environments a startup script, **QSAS**, is provided. Each user should copy the script **QSAS** from the **QSAS_home/bin** directory into their own bin directory (or other directory in their path) and edit it to set the user specific environments.

The following environment variables in the script may be edited:-

DISPLAY This environment variable should be set to *hostname:0.0* where *hostname* is the name (or IP address) of the host on which the windows are to be displayed

(the X terminal or workstation at which the user is sitting).

Q_SAVE_RESTORE_DIR This environment variable should be set to `/home/UserName/tmp` where **UserName** is the name of the user's home directory. A different location may be specified, but the user must have write permission, and the location should be unique for each user to avoid confusion between different sessions.

This script should have execute permission set:

```
% chmod u+x QSAS
```

QSAS may now be started by giving the command

```
% QSAS
```

at the system prompt, or

```
% QSAS hostname
```

to set the DISPLAY environment to a different machine from that identified in the script.

If you get the message "QSAS: command not found" the first time you attempt to run QSAS, type "rehash" and try again.

To display the QSAS window on a machine from that running QSAS the machine you are sitting at must allow the remotehost to display the X-Window on it. At the prompt type

```
% xhost +remotehost
```

where remotehost is the name or domain address of the remote machine that QSAS is running on.

Alternatively, you can use ssh to establish communications with the remote host, in which case the DISPLAY environment should be commented out in the QSAS script and you do not need to set xhost on your terminal. To use ssh, type

```
% ssh -lusername -X remotehost
```

where the -l flag is only needed if your user id is different on the remote and local machines, and the -X flag is only required if X-Forwarding is not provided by default.

A sample qsas session file is included in the distribution for new users to experiment with. From the QSAS main window go to File and select **Restore Session**. In the file dialog window navigate to the **QSAS.2.2** directory and select SampleSession.qss which is to be found in the directory tmp. This will restore qsas to a session with data already loaded and some of the other windows in an active state showing how they work.

6 Compiling local Plug-ins

A script buildPlugin is provided in the bin directory to assist in building local plug-ins. All QSAS supplied plug-ins are built automatically with the distribution.

Copy this buildPlugin script to the folder containing your plug-in source code and edit the environments in it in the same way as you edited the build script. Copy a Makefile from one of the QSAS supplied plug-ins in `/src/Ext` into your plug-in directory and edit it to reflect the names of your plug-in source files and .so library to be created. Typing buildPlugin will build and install your plug-in shareable object library. The QSAS menu Plug-ins has an item **Refresh Menu** which will render any plug-in visible to QSAS after its .qtpl file has been installed in the qtpl/Analysis or qtpl/Geophysics directories.

7 Troubleshooting

This section lists problems that may arise from installation at new sites. The areas of concern are divided into subsections according to functionality within QSAS that exhibit the symptoms.

7.1 Launching qsas

7.1.1 Execution error

Symptoms: System error message “Cannot find library...”

Cause (1): This is likely if the LD_LIBRARY_PATH - set in QSAS - does not contain the correct path to the system libraries.

Action: Set the environment “setenv LD_DEBUG files” and try QSAS again. This will result in full library version and path information being written to stdout and may help locate which libraries are at fault.

Cause (2): This may also result if the libraries linked when the executable was built at IC differ from those on your local system.

Action: Some externally provided libraries are included as binaries in **QSAS_home**/lib/syslib and adding this path to the start of the LD_LIBRARY_PATH in **QSAS** may allow qsas to execute. To help diagnose the library versions loaded change the line in the QSAS script (temporarily) from “\$QSAS_HOME/lib/qsas” to “ldd \$QSAS_HOME/lib/qsas”. This will show the libraries being loaded and their version numbers. It may be necessary to rename the system libraries supplied in \$QSAS_HOME/lib/syslib to that needed.

7.1.2 Execution error

Symptoms: QSAS crashes immediately with a segmentation fault.

Cause (1): This is almost certainly a library incompatibility between IC and your local system as above.

Action: Build an executable locally from the autoconf source release if the preceding tip does not work.

7.1.3 qsas file read/write errors

Symptoms: Files not owned by installer or accessible by group.

Cause (1): If installation is performed by root the original file ownership is restored on tar extraction.

Action: If installation is performed by root the **-o** option should be included in tar extraction to ensure local file ownership. Files will still be owned by root, so installation by a user within the same group as the science users is recommended.

7.1.4 No QSAS GUI window

Symptoms: Unable to connect to display host

Cause (1): This may occur if the host running qsas does not have permission to display on the host used for display (where you first login).

Action: Add the remote machine(s) to your xhost settings by
% xhost *ip_address_of_remote_host*

Cause (2): This may occur if the DISPLAY environment variable is not set correctly.

Action: Check your personal copy of QSAS script for DISPLAY environment setting. If the host running qsas does not know the name of your display host you may need to set the full name, *e.g.* cdhf4.bnsc.ral.ac.uk or the ip address in place of machine name.

7.2 Data Access

7.2.1 QDS cannot get data

Symptoms: .dcm file is not available

Cause (1): qsas must first create a .dcm file for the data held.

Action: Under **Data Files : Data Selection** choose **Database : Create Database Entries** and navigate file hierarchy to select data files you wish to add to the database using left button click and left button shift-click to select contiguous files.

7.2.2 QDS cannot get data

Symptoms: .dcm file loaded but data is not available for selected time

Cause (1): This may result if the time interval set does not contain any data in the configured database.

Action: Set the time selection to start 1990 and finish 2002 and try again.

7.2.3 qsas crashes on data access

Symptoms: qsas crashes at the time a cdf data file is opened.

Cause (1): The **cdflib** functions provided by CDF will crash if the cdf file is incomplete. This may occur if an ftp on that file fails or a write error (*e.g.* full disc) occurs when extracting the data from the tar file.

Action: Delete the offending cdf file(s). This is a bug in cdf and cannot be circumvented.

7.3 Plug-ins

7.3.1 Cannot find Plug-in entry point

Symptoms: Plug-ins always return failure.

Cause (1): The plugin must be declared external.

Action: If the entry name in the error message is a long string with your plugin name bedded in it, this is the same problem as migration from version 1.6 above.

7.3.2 Plugin fails to parse or run.

Symptoms: A plugin that used to work now doesn't.

Cause (1): The qtpl parser has been replaced, and compilers are becoming more strict.

Action: The new qtpl parser is more tolerant than the old one, so this is unlikely, however if a qtpl doesn't parse the console and QSAS window will now hold debug information from the parser. Stricter compilers mean that warnings can become errors, so code may need tidying up.

7.3.3 Locally written plug-in requires debugging

Symptoms: Since QSAS supports locally written plug-ins it is convenient to be able to debug directly using qsas rather than writing a test bench for each plug-in.

Cause (1): QSAS executables are shipped with the debug symbols stripped out in order to make smaller executables.

Action: An executable qsas binary which will run with the xxgdb debugger can be fetched separately from the IC web site. Note this file contains only the actual qsas executable which should be placed in **QSAS_HOME/lib**

7.4 Reporting problems

In the case of a new installation problem not covered above, or uncertainty in installation procedures contact csc-support-dl@imperial.ac.uk.

The QSAS signal and slot tracking debug facility can be turned on by un-commenting the line in the **QSAS** script

```
setenv QSAS_DEBUG "ON"
```

It will help CSC staff to locate a problem if the terminal output generated (or last couple of pages) are forwarded with the bug report.

The Save Session utility can also be used to assist support staff. Run the session until just before failure, save session, then note actions performed until qsas failure. This information, together with a tar file containing the saved session files will assist us in reproducing the problem.

When reporting a problem, please provide information on the operating system being used and its version, the nature of the failure and, if possible, a sequence of operations that led to the failure and any output generated. We will also need to know if this installation was from the binary executable distributed or built from source.