Star Traders: A Game of Interstellar Trading

Star Traders is written in the C99 programming language and uses Autoconf and Automake to handle compilation and installation. Assuming you have the needed tools, all you should need to do is run the following commands from the source directory:

```
./configure
make
make install
```

The first two commands may be run as an ordinary user; the last may need to be done as the system administrator (root).

Contents

Prerequisites 1

Installation 2

Tested Systems 3

Git Repository 5

For Translators 6

Prerequisites

Star Traders requires the following components for successful compilation and installation:

1. A working C compiler conforming to ISO/IEC 9899:1999 (also known as C99) or later. Any recent version of the GNU Compiler Collection (GCC) or the Clang LLVM Compiler is more than adequate.
2. An operating system ideally conforming to ISO/IEC 9945-1:2008 (POSIX) or to the Open Group Single UNIX Specification version 4 or later. In short, any modern Unix or Unix-like system like Linux almost certainly qualifies.

   In actual fact, Star Traders uses the GNU Portability Library, so many older systems may also work without modification.
3. A working X/Open Curses-compatible library, such as Ncurses. Ncurses is preferred over system-native libraries, if present. Locales with multibyte character sequences (such as UTF-8) require a wide-character version of Curses, such as NcursesW, to work correctly.
4. The GNU Gettext library, version 0.21 or later, to allow the game to use languages other than English; this is also called Native Language Support. If you do not have this library (and do not wish to install it), you may pass `--disable-nls` to the configure script.
5. The GNU `libiconv` library for supporting multiple character encodings, if required by the GNU Gettext library. This is not needed on systems with the GNU C Library (`glibc`) version 2.2 or later, or on macOS 10.3 or newer.
6. Development libraries and header files for all of the above. On many systems, these files are part of `XXX-dev` or `XXX-devel` packages.
7. The GNU Perfect Hash Function Generator, `gperf`. This utility program may be required for parts of the GNU Portability Library.

Installation

The installation of Star Traders can be broken down into three main steps: configuration, compilation and the installation proper.

The first step is configuring the package for your compiler and operating system environment. As Star Traders uses Autoconf and Automake, all you need to do in most cases is run `./configure` from the top-level directory of the Star Traders source tree. The configure script understands all of the usual Autoconf options; these are explained in detail in the Autoconf manual.

This version of the configure script understands the following additional command line options:

`--disable-nls`	Don't use Native Language Support. Star Traders will only show untranslated US English text and only accept US ASCII keyboard input.
`--with-libintl-prefix=DIR`	Find the GNU Gettext library installed in the *DIR/lib* and *DIR/include* directories. This option is needed if your library is not installed in */usr/lib* and */usr/include* (or, more precisely, if the C compiler cannot find the library "`.so`" or "`.a`" archive file using the standard "`-l`" command line parameter or the relevant header files using standard `#include` directives).
`--with-libiconv-prefix=DIR`	Find the GNU `libiconv` library installed in the *DIR/lib* and *DIR/include* directories. This option is needed if the GNU Gettext library requires `libiconv` on your system, and that library is not installed in */usr/lib* and */usr/include*.
`--with-ncurses`	Force the use of Ncurses over the system's Curses library. In other words, do not search for a native Curses library at all.
`--with-ncursesw`	Force the use of the NcursesW library with wide-character support. If NcursesW cannot be found, abort the `configure` script.
`--without-ncursesw`	Don't use the NcursesW library. This will prevent non-8-bit character encodings like UTF-8 from working correctly and is thus not recommended.
`--without-ncurses`	Don't use the Ncurses library: use either NcursesW (unless `--without-ncursesw` is also specified) or the system's normal Curses library. This option is not recommended.
`--disable-assert`	Turn off all debugging `assert()` statements.

By default, `configure` uses */usr/local* as the top-level (prefix) install directory. You can change this by specifying `--prefix=`*DIR* to use *DIR* instead. For example, you can use a directory in your own home directory by specifying something like:

```
./configure --prefix=$HOME/opt/trader
```

You may also specify certain configuration and/or compilation variables on the command line to override choices made by `configure`. For example, you can specify the compiler flags to use by passing the `CFLAGS` variable:

```
./configure CFLAGS="-g -O2 -Wall"
```

The `configure` script has many other options. You may obtain a list of these by running:

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

You can also run `configure` in a separate build-only directory tree. This feature requires GNU Make and allows you to keep the source code tree from being modified by the compilation process. To use this option, create a separate *build* directory, then run `configure`. For example, if you placed the Star Traders source code tree in *$HOME/src/trader-7.20*, you could run something like:

```
mkdir $HOME/build/trader-build-7.20
cd $HOME/build/trader-build-7.20
$HOME/src/trader-7.20/configure
```

Once again, the Autoconf manual describes these options (and many others).

Once the package has been configured, you can type `make` to compile it, then `make install` to install it. You can specify the following command lines, amongst others:

```
make all
make install
make clean
make distclean
make uninstall
```

The command `make all` does the same thing as running `make` by itself: compile the package source code into an executable.

Running `make install` copies the executable program and all associated data and documentation files to those directories specified during configuration. If any of these directories require system administrator access privileges for writing, you will need to run `make install` as system administrator (root).

If you like, you can specify the `DESTDIR` variable to copy all installation files to a temporary location before installing them later. For example, if the prefix directory is */usr/local*, typing:

```
make install DESTDIR=/tmp/trader-install
```

will copy the final program `trader` to */tmp/trader-install/usr/local/bin*, the manual page to */tmp/trader-install/usr/local/share/man/man6* and so on.

The `make clean` command will remove most build-generated files, such as object files generated by the compiler, from the source code or build directory. Running `make distclean` will do the same, but will remove the Makefiles generated by `configure` as well. In other words, if you run `make distclean`, you will need to rerun `configure` if you would like to recompile Star Traders at a later date.

Finally, `make uninstall` will remove the executable program `trader` and associated data and documentation files from their final installation location. This assumes, of course, that you have *not* run `make distclean` to remove the Makefiles that know the path to which those files were installed!

Tested Systems

The following operating systems and compilers have been successfully tested with this version of Star Traders:

Linux distribution	Arch	Glibc	Compiler
Debian GNU/Linux Unstable (Sid)	x86_64	2.37	GNU C Compiler 13.2.0
Debian GNU/Linux Unstable (Sid)	x86_64	2.37	Clang (LLVM) 16.0.6
Debian GNU/Linux 12.4 (Bookworm)	x86_64	2.36	GNU C Compiler 12.2.0
Debian GNU/Linux 12.4 (Bookworm)	x86_64	2.36	Clang (LLVM) 14.0.6
Ubuntu 23.10 (Mantic Minotaur)	x86_64	2.38	GNU C Compiler 13.2.0
Ubuntu 23.10 (Mantic Minotaur)	x86_64	2.38	Clang (LLVM) 16.0.6
Ubuntu 22.04.3 LTS (Jammy Jellyfish)	x86_64	2.35	GNU C Compiler 11.4.0
Ubuntu 22.04.3 LTS (Jammy Jellyfish)	x86_64	2.35	Clang (LLVM) 14.0.0
Ubuntu 20.04.6 LTS (Focal Fossa)	x86_64	2.31	GNU C Compiler 9.4.0
Ubuntu 20.04.6 LTS (Focal Fossa)	x86_64	2.31	Clang (LLVM) 10.0.0
Ubuntu 18.04.6 LTS (Bionic Beaver)	x86_64	2.27	GNU C Compiler 7.5.0
Ubuntu 18.04.6 LTS (Bionic Beaver)	x86_64	2.27	Clang (LLVM) 6.0
Fedora Linux 39	x86_64	2.38	GNU C Compiler 13.2.1
Red Hat Enterprise Linux 9.3	x86_64	2.34	GNU C Compiler 11.4.1
Red Hat Enterprise Linux 8.9	x86_64	2.28	GNU C Compiler 8.5.0
Red Hat Enterprise Linux 7.9	x86_64	2.17	GNU C Compiler 4.8.5
Red Hat Enterprise Linux 6.10	x86_64	2.12	GNU C Compiler 4.4.7
CentOS Stream 9	x86_64	2.34	GNU C Compiler 11.4.1
CentOS Stream 8	x86_64	2.28	GNU C Compiler 8.5.0
CentOS 7.9.2009	x86_64	2.17	GNU C Compiler 4.8.5
Rocky Linux 8.8	x86_64	2.28	GNU C Compiler 8.5.0
Rocky Linux 8.8	x86_64	2.28	Intel oneAPI C 2023.2.0

Rocky Linux 8.8	x86_64	2.28	Intel C Classic 2021.10.0
openSUSE Leap 15.5	x86_64	2.31	GNU C Compiler 7.5.0
Arch Linux (2024.01.05)	x86_64	2.38	GNU C Compiler 13.2.1

Operating system	Arch	Compiler	Notes
FreeBSD 14.0	x86_64	Clang (LLVM) 16.0.6	[1]
NetBSD 9.3	x86_64	GNU C Compiler 7.5.0	[2]
macOS 12.6	x86_64	Apple Clang (LLVM) 14.0.0	[3]
macOS 12.6	x86_64	Apple Clang (LLVM) 14.0.0	[4]
Solaris 11.4.42 (x86_64)	x86_64	GNU C Compiler 11.2.0	[5]
Solaris 11.4.42 (x86_64)	i386	GNU C Compiler 11.2.0	[6]
Solaris 11.4.42 (x86_64)	x86_64	Oracle Developer Studio 12.6	[7]
Solaris 11.4.42 (x86_64)	i386	Oracle Developer Studio 12.6	[8]
Haiku R1/beta4	x86_64	GNU C Compiler 13.2.0	
Cygwin 3.4.10 (Windows 10)	x86_64	GNU C Compiler 11.4.0	

The following systems are known *not* to work at the current time; this list is almost certainly not exhaustive:

Operating system	Arch	Compiler	Notes
Linux (any)	x86_64	NVIDIA HPC SDK 23.11	[9]
OpenBSD 7.4	x86_64	Clang (LLVM) 13.0.0	[10]

Git Repository

You can always download the latest version of Star Traders directly from the Git repository on The ZAP Group Australia server:

```
git clone git://git.zap.org.au/data/git/trader.git
```

Released versions of Star Traders include all scripts and files needed for installation. If you are cloning the source code from the Git repository, however, you will need to update these files yourself. You will need the following additional tools installed on your system to do so:

1. Autoconf v2.71 or later
2. Automake v1.16 or later
3. pkgconf v0.9.0 or later, or pkg-config v0.29 or later
4. GNU Portability Library

The GNU Portability Library may be installed by retrieving the latest Gnulib source code from the Git repository:

```
git clone git://git.savannah.gnu.org/gnulib.git
```

Once you have these tools, change to the Star Traders source code tree and type:

```
PATH=${PATH}:/path/to/gnulib-tool ./build-aux/bootstrap
```

where `/path/to/gnulib-tool` is, of course, the directory containing the Gnulib `gnulib-tool` script. You should be ready to run `./configure && make && make install` now.

For Translators

Thank you for even considering to translate Star Traders into your native language! You may use either a released version of Star Traders, or an unreleased one, as discussed in the Git Repository section above. In either case, you may find the following workflow useful.

First, run `./build-aux/bootstrap` if needed (only for unreleased versions of Star Traders).

Next, configure and install Star Traders into your home directory:

```
./configure --prefix=$HOME/opt/trader
make
make install
```

If you are adding a new translation, add its GNU Gettext language code to the file *po/LINGUAS*, then create the template file for that language ("`zz`" is used here):

```
(cd po; msginit --locale=zz --width=132)
```

Now, modify the PO file for your language using your favourite editor or translation tool. Please note that the generated PO file has extensive documentation in its translator comments. If anything is unclear, please feel free to ask the author and maintainer; contact details are available in the *README* file.

To test your PO file, compile and run Star Traders (replace "`zz`" with your language code, of course):

```
make && make -C po zz.gmo && make install
LANGUAGE=zz $HOME/opt/trader/bin/trader
```

The `make -C po zz.gmo` forces the rebuilding of the GMO output file; the `LANGUAGE=zz` parameter sets the language of the messages to use.

This process of editing and testing the PO file can be done iteratively, of course: make a change, recompile, run the program to see the changes, repeat as needed.

Once you have finished your translation, please submit the PO file to the Translation Project (TP). See the TP Star Traders web page or read their Translators and the TP page for additional information.

To clean up your install directory, simply run:

```
rm -fr $HOME/opt/trader
```

By the way, as mentioned in the translator comments, formatting the help text is probably the most complicated and tedious part of translating Star Traders. The author and maintainer of this game is more than happy to help you with this task: if you are able to provide a translation, even if it is not

formatted correctly, the maintainer will perform the necessary adjustments for word-wrapping and justification.

---

1 FreeBSD with the `gettext`, `gettext-runtime`, `gettext-tools` and `libiconv` binary packages installed with `pkg`(1), using:

```
./configure --with-libiconv-prefix=/usr/local \
            --with-libintl-prefix=/usr/local
```

2 NetBSD with the `gettext`, `libiconv` and `ncursesw` packages installed with `pkgin`(1), using:

```
./configure LDFLAGS=-L/usr/pkg/lib
```

Note that current versions of NetBSD have a known bug in their `strfmon()` function: if the locale is not `C` or `C.UTF-8`, amounts may be displayed incorrectly. For example, under the `en_US.UTF-8` locale, an amount of `$6000` is displayed as `$6,000.0,000.00` instead of `$6,000.00`.

If the current locale is `C`, game loads and saves may fail with the error message `trader: iconv_open: Invalid argument`. The `C.UTF-8` locale works correctly.

3 macOS with Xcode command line tools, with the `gettext` and `pkg-config` packages installed with Homebrew, using:

```
./configure
```

4 macOS with Xcode command line tools, with the `gettext`, `ncurses` and `pkg-config` packages installed with Homebrew, using:

```
./configure PKG_CONFIG_PATH=/usr/local/opt/ncurses/lib/pkgconfig
```

5 Solaris, using:

```
export PKG_CONFIG_PATH=/usr/lib/64/pkgconfig
./configure CC="gcc -m64"
```

6 Solaris, using:

```
export PKG_CONFIG_PATH=/usr/lib/32/pkgconfig
./configure CC="gcc -m32"
```

7 Solaris with Oracle Developer Studio, using:

```
export PKG_CONFIG_PATH=/usr/lib/64/pkgconfig
./configure CC="/opt/developerstudio12.6/bin/cc -m64"
```

8 Solaris with Oracle Developer Studio, using:

```
export PKG_CONFIG_PATH=/usr/lib/32/pkgconfig
./configure CC="/opt/developerstudio12.6/bin/cc -m32"
```

9 The NVIDIA HPC SDK 23.11 compiler (and possibly earlier versions) does not work due to a bug in `nvc -E` preprocessor output. See the bug report for further details.

10 The OpenBSD C library does not include `<monetary.h>` nor its associated functions, particularly `strfmon()`.