

# How to install HDF5 1.6.6 for Suse Linux 9.3

Andreas Maier

August 7, 2008

## 1 Introduction

We describe how to compile and install HDF5-1.6.6 using the Intel C++ and Fortran Compilers V10.1 on a Suse Linux 9.3 with 32bit CPU.

## 2 Installing the Intel-Compilers Version 10.1 for C/C++ and Fortran90

On the Intel Web pages for the non-commercial unsupported version of their compilers

<http://www.intel.com/cd/software/products/asm-na/eng/340679.htm>

you should choose to download the 32 bit versions of the Intel compilers:

```
l_cc_p_10.1.008_ia32.tar.gz  
l_fc_p_10.1.008_ia32.tar.gz
```

You will also receive one email for each of the compilers including your serial number and licence file \*.lic. Save the license files to some directory.

Use `tar zxvf [filename]` to decompress and unpack the \*.tar.gz-files to a temporary directory.

Install the compilers by calling `./install.sh` in the temporary directory directory of each compiler. It is important to use `./`, otherwise one might call a different `install` existing in the `PATH`.

It is best to install first the C/C++-compiler (`icc`, `icpc`), then the Intel debugger (`idb`) and then the Fortran compiler (`ifort`). You should not install the Intel debugger again from the Fortran compiler installation routine.

### 3 Compiling and installing SZIP 2.1

---

The installation should run without major problems. One tip: Instead of providing the path to the license files during the installation you can just give the serial numbers you received via email.

After installing the two compilers to some directory [INSTALL\_PATH\_ICC], [INSTALL\_PATH\_IDB] and [INSTALL\_PATH\_IFORT] you should copy the license files for the Fortran compiler and the C/C++ compiler to the directories

```
[INSTALL_PATH_ICC]/licenses
[INSTALL_PATH_IFORT]/licenses
```

respectively. Check the permissions of the \*.lic-files and change them if other users should be able to use the compiler as well:

```
> chmod a+r [INSTALL_PATH_ICC]/licences/*.lic
> chmod a+r [INSTALL_PATH_IFORT]/licences/*.lic
```

One has to add the compilers to the PATH. Therefore one can just copy the content of the scripts [INSTALL\_PATH\_ICC]/bin/iccvars.sh and [INSTALL\_PATH\_IFORT]bin/ifortvars.sh to the file .bashrc in the home directory of each user or add the following lines to the file /etc/bash.bashrc.local (not /etc/bash.bashrc, because changes there will be lost during a system upgrade):

```
. [INSTALL_PATH_IFORT]/bin/ifortvars.sh
. [INSTALL_PATH_ICC]/bin/iccvars.sh
. [INSTALL_PATH_IDB]/bin/idbvars.sh
```

This ensures, that every user on the local computer has the compilers in his PATH.

## 3 Compiling and installing SZIP 2.1

For additional SZIP support in HDF5 you need the SZIP source code V2.1 from <ftp://ftp.hdfgroup.org/lib-external/szip/2.1/src/szip-2.1.tar.gz>

Decompress the archive

```
> tar zxvf szip-2.1.tar.gz
> cd szip-2.1
```

and in the SZIP directory build SZIP by doing the following steps

```
> ./configure --prefix=[INSTALL_PATH_SZIP]
  --enable-static=yes --disable-shared
> make
> make check
> make install
```

It is very important to use the option --disable-shared, otherwise the SZIP libraries cannot(!) be used to build HDF5.

---

## 4 Compiling and installing HDF5-1.6.6

Since version 1.6.6<sup>1</sup> the developers of HDF5 provide precompiled binaries for Linux Kernel 2.6.9:

<http://hdf.ncsa.uiuc.edu/HDF5/release/obtain5.html>

They use the `gcc` and `g95`-compiler and if that's ok for you just try these binaries. But if you will compile your application with an Intel compiler anyway it is probably better to compile HDF5 with the Intel compilers.

To do this you need the source code for HDF5 available from

<ftp://ftp.hdfgroup.org/HDF5/current/src/hdf5-1.6.6.tar.gz>

Decompress the archive

```
> tar zxvf hdf5-1.6.6.tar.gz
> cd hdf5-1.6.6
```

Before configuring and compiling you have to make sure, that you do not have your environment variable `LD_LIBRARY_PATH` pointing to some old version of HDF5

```
> export LD_LIBRARY_PATH=[PATHS_WITHOUT_OLD_HDF5]
```

because otherwise the checks during installation will fail. Then to use the Intel compiler you have to set the environment variables

```
> export CC=icc
> export CXX=icpc
> export FC=ifort
```

After doing this we are ready to build and install HDF5 (with Fortran and C++-Headers and SZIP support)

```
> ./configure --prefix=[INSTALL_PATH_HDF5]
  --with-szip=[INSTALL_PATH_SZIP] --enable-fortran
  --enable-cxx
> make
> make check
> make install
```

Do not(!) use the procedure suggested in the manual to HDF5

---

<sup>1</sup>In HDF5 V1.6.4 some APIs in HDF5 have changed and therefore some programs are incompatible with the newest versions of HDF5. If this is the case for your application, you have to install version V1.6.3 of HDF5 (see my other manual at <http://www.astro.uni-wuerzburg.de/~asmaier/documents/hdf5old.pdf>).

```
> ./configure ...
> make check >& check.log
> make install
```

as `make check` will fail for some unknown reason.

Because HDF5 is compiled as a shared library you have to make sure that executables find the library during runtime and you should add the following line to your `.bashrc`

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:[INSTALL_PATH_HDF5]/lib
```

This makes sure that the HDF5 libraries are in your library search path.

One last tip: If you want to compile your program using the C/C++ or the Fortran header files be sure not to forget the options `-lhdf5`, `-lhdf5_cpp` or `-lhdf5_fortran` respectively. If compilation fails due to missing ZLIB or SZLIB also use the options `-lz` and `-lsz`<sup>2</sup>

---

<sup>2</sup>If the linker cannot find the zlib on Suse 9.3, you have to install the package `zlib-devel` with Yast2.