

Libraries

December 17, 2020

Contents

1 ITM Libraries

1

The ITM provides a number of libraries specific to ITM-related concepts and tasks. Furthermore a number of standard libraries are available for code development. This page gives an overview of what's available and how to use it.

1 ITM Libraries

Libraries developed by the ITM are made available on the ITM platform in standardized directory layout. The root for this directory hierarchy is currently placed at `$ITMLIBDIR`. However, please do not hard-code this path into your Makefiles and build systems. An environment variable `ITMLIBDIR` holding this value is automatically defined in the user environment.

The library files are organized as follows:

- `$ITMLIBDIR` (`$ITMLIBDIR`)
 - Library name (e.g. itmconstants)
 - * Data version (e.g. 4.10a)
 - **include** - compiler-independent include files (e.g. C header files)
 - **lib**
 - **OBJECTCODE (see below)** - compiler-dependent files (*.a, *.so library archive files (e.g. libitmaggd.a), *.mod Fortran module files (e.g. itm.types.mod))
 - **pythonX.Y** - Python modules for Python version X.Y (see below)
 - compiler-independent library files (e.g. libUALLowLevel.a)
 - **libdebug** - same as lib, but with debug symbols and no optimization

In the hierarchy, different compilers and compiler versions are distinguished with an OBJECTCODE identifier. Currently, the following values are used:

- **amd64.pgi.10** - Portland Group, Inc. compilers (C, C++, Fortran), Release 10
- **amd64.g95.0.92** - g95 compiler, version 0.92
- **amd64.gfortran.4.7** - gfortran compiler, version 4.7

However, not all libraries are built for all compilers depending on compatibility.

The full draft for the ITM standard directory layout is described in this document: [ITM standard directory layout](#)¹.

last update: 2013-09-12 by dpc

¹https://www.efda-itm.eu/ITM/imports/isip/public/ITM_Library_Directory_Layout.pdf