

- [MODROOT](#)
 - [MODROOT_WS](#)
 - [MODROOT_LCU](#)
 - [MODROOT_WS_LCU](#)
 - [Files and Directory Description](#)
- [ACSRROOT](#)
- [ACSDATA](#)
 - [Files and Directory Description](#)
- [INTROOT/INTLIST](#)
 - [INTROOT](#)
 - [INTLIST](#)

The ACS directory structure is very straightforward, but it may not be very intuitive to newcomers. There are a couple of concepts that we need to consider:

- MODROOT_WS and MODROOT_LCU, MODROOT_WS_LCU
- ACSRROOT
- ACSDATA
- INTROOT and INTLIST

All three variants of MODROOT represent a very similar scenario. They create a module directory, either for WorkStation (MODROOT_WS), for Local Control Unit (LCU) (MODROOT_LCU) or for both (MODROOT_WS_LCU). Executing each bears a similar structure:

MODROOT_WS

```
getTemplateForDirectory MODROOT_WS wsModule
```

```
acs almamgr:~ 30449 > tree wsModule/
wsModule/
bin
ChangeLog
config
  CDB
    schemas
doc
idl
include
lib
  ACScomponents
  endorsed
  python
    site-packages
LOGS
man
  man1
  man2
  man3
  man4
  man5
  man6
  man7
  man8
  man1
  mann
object
rtai
src
  Makefile
test
```

MODROOT_LCU

```
getTemplateForDirectory MODROOT_WS lcuModule
```

```
acs almamgr:~ 30451 > tree lcuModule
lcuModule
bin
ChangeLog
doc
idl
include
lib
    ACScomponents
    endorsed
    python
    site-packages
LOGS
man
    man1
    man2
    man3
    man4
    man5
    man6
    man7
    man8
    man1
    mann
object
src
    Makefile
test
```

MODROOT_WS_LCU

```
getTemplateForDirectory MODROOT_WS_LCU module
```

```

acs almangr:~ 30459 > tree module
module/
  ChangeLog
  lcu
    bin
    doc
    idl
    include
    lib
      ACScomponents
      endorsed
      python
      site-packages
    LOGS
  man
    man1
    man2
    man3
    man4
    man5
    man6
    man7
    man8
    man1
    mann
  object
  src
    Makefile
  test
ws
  bin
  config
    CDB
    schemas
  doc
  idl
  include
  lib
    ACScomponents
    endorsed
    python
    site-packages
  LOGS
  man
    man1
    man2
    man3
    man4
    man5
    man6
    man7
    man8
    man1
    mann
  object
  src
    Makefile
  test

```

Files and Directory Description

As we can see, there are several files and directories created under each of the scenarios:

- bin: Here we place the scripts and binaries that get executed. This could be added to PATH environment variable
- Changelog: A file that could be used to register changes of the module. We make use of JIRA and "git log" for this, but this could be used as an additional place for this information.

- config: Directory for configurations.
 - config/CDB: Directory for module's CDB configurations. Could be used with ACS_CDB environment variable to configure a CDB for testing the module
 - config/CDB/schemas: The XML Schema (XSD) files are placed in this directory
- doc: Directory to place documentation
- idl: Directory for IDL (.idl, .midl) files, XML error definitions, XML Schema binding file, etc.
- include: Directory for C++ headers, inline and template files
- lib: Directory for C++ libraries and Java JAR files. Could be used in LD_LIBRARY_PATH (C++ libraries) and in CLASSPATH (Jar files) to make use of the module's libraries and jar files
 - lib/ACSComponents: Directory for Java JAR files associated with components
 - lib/endorsed: Directory for endorsed JAR files (No longer relevant since Java 11 disabled the use of endorsed JARs)
 - lib/python: Directory for Python code
 - lib/python/site-packages: Python site-packages are placed here. Could be used with PYTHONPATH environment variable to make use of the Python code and cached code (.pyc)
- LOGS: Directory where logs can be placed
- man: Man documentation in Unix/Linux style could be placed here. See more at <https://linux.die.net/man/> page.
 - man/man<n>: Look at the link above
 - man/man1: Look at the link above
 - man/mann: Look at the link above
- object: Directory for temporary files such as C++ compiled objects, code-generated source files, etc.
- rtai: Directory for Real Time Application Interface (RTAI) and other compiled kernel modules to be installed
- src: Directory for source files
 - src/Makefile: The main file to define the targets to be compiled, installed and deployed by the module
- test: Directory that contains all the necessary files for testing. Source code, Makefile, TAT configuration files, reference files, etc.

The ACSROOT directory is considered an "installation area". When you compile a module and call "make install", ACSROOT is the default destination area for ACS modules' installation.

```
acs almamgr:~ 31443 > getTemplateForDirectory ACSROOT acs.root
```

```

acs almangr:~ 31445 > tree acs.root/
acs.root/
  ALARMS
    HELP
  app-defaults
  bin
  bitmaps
  CDT
  config
    CDB
      schemas
  ERRORS
    HELP
  idl
  include
  lib
    ACScomponents
    endorsed
    python
      site-packages
  LOGS
  man
    man1
    man2
    man3
    man4
    man5
    man6
    man7
    man8
    manl
    mann
  rtai
  sounds
  templates
  vw
    bin
      MC68000
      MC68040
      PPC604
    CIT
    idl
    include
    lib
      ACScomponents
      endorsed
      MC68000
      MC68040
      PPC604
      python
        site-packages
    man
      man1
      man2
      man3
      man4
      man5
      man6
      man7
      man8
      manl
      mann

60 directories, 0 files

```

The directory structure is very similar to MODROOT, bin, config, idl, include, lib, LOGS, man and rtai are equivalent for the MODROOT definitions, although only the targets selected for installation are finally placed there. Changelog, src, doc, object and test are not found in the installation areas. Other directories are as follows:

- ALARMS and ALARMS/HELP: Alarm definition documentation (Not used)
- app-defaults: Not used.
- bitmaps: Not used.
- CDT: Not used.
- ERRORS and ERRORS/HELP: Error definition documentation (Not used)
- sounds: Not used
- templates: Templates, such as the ones used for getTemplateForDirectory command are placed here
- vw: VxWorks. Not used in most projects. Used at APEX observatory.

ACSDATA is a place associated with runtime configurations and output of data.

```
getTemplateForDirectory ACSDATA acs.data
```

```
acs almangr:~ 30451 > tree acs.data
acs.data
  config
  dumps
  ENVIRONMENTS
  logs
  tmp

5 directories, 0 files
```

Files and Directory Description

- config: Configuration files, including CDB, schemas and other files
- dumps: Core dumps, heap dumps, etc. can be placed in this directory
- ENVIRONMENTS: Not used
- logs: Runtime logs can be placed in this directory
- tmp: Runtime temporary data is placed here

INTROOT is an installation area with the same structure as ACSROOT, but the main difference is that it allows to install code without interfering with the ACS (or your project installation). It is very handy for development and experimental tests and also as a patching system, specially when used in combination with INTLIST. INTLIST is just a list of INTROOTS, which have an order of prioritization which will be used to choose binaries, libraries, jars, python code, etc. from one place or another. Everything that is in two or more INTROOTS gets hidden for the lower priority one by the one with highest priority.

INTROOT

```
getTemplateForDirectory INTROOT introot
```

```
acs almangr:~ 30463 > tree introot
```

```
introot
  ALARMS
    HELP
  app-defaults
  bin
  bitmaps
  CDT
  config
    CDB
      schemas
  ERRORS
    HELP
  idl
  include
  lib
    ACScomponents
    endorsed
    python
      site-packages
  LOGS
  man
    man1
    man2
    man3
    man4
    man5
    man6
    man7
    man8
    manl
    mann
  responsible
  rtai
  sounds
  Sources
  templates
  vw
    bin
      MC68000
      MC68040
      PPC604
    CIT
    idl
    include
    lib
      ACScomponents
      endorsed
      MC68000
      MC68040
      PPC604
      python
        site-packages
    man
      man1
      man2
      man3
      man4
      man5
      man6
      man7
      man8
      manl
      mann
    Sources
```

```
62 directories, 1 file
```

The same directory structure as ACSROOT is present. Here you see one more:

- Sources: Here, a copy of src/include directories get installed along with your build, allowing you to analyze exact pieces of code used to produce the build. This directory is also installed to ACSROOT, but the getTemplateForDirectory is actually ignoring the creation of the Sources directory.

INTLIST

The INTLIST is just an environment variable used by the ACS .bash_profile.acs script to configure several INTROOTS in an order of priority.

```
export INTLIST=~/introot1::~/introot2::~/introot3::~/introot4
```

In the example above, it will have four INTROOTS configured in parallel, where ~/introot1 has the highest priority and ~/introot4 the lowest. This means that every binary or script offered by ~/introot1 will be used first, and things from ~/introot4 will only be used if they're not shadowed by a different version provided in the other 3 INTROOTS.