ExtProd

buildPython

- Several variables make reference to python version to be used.
- TODO: Includes removal of configuration options for TCL/TK. Are we still using TCL/TK in this new release? If not, then these options should be leaved commented.
- Added here the symbolic link from python to python 3 (this is discussed down below)

buildPyModules

- use the pip that comes with python3.
- updates this pip installation to latest version before installing other modules
- TODO: Create a small scripts that captures errors during installation of pip acs.req. If one module fails, it just aborts and the build script never realizes if there was an error or not.

Several errors installing modules. Most of them due versions of python modules not compatible with python 3.

- distribute: AttributeError: module 'importlib._bootstrap' has no attribute 'SourceFileLoader'
  - Is deprecated, and superseed by setuptools.
  - In pip, there is a wrapper package that install setuptools instead.
  - Deleted distribute from acs.req and install latest version of setupttols instead.
- gnuplot-py: Missing module StringIO and cStringIO
  - Stopped being developed in 2008
  - Two possible replacements: https://github.com/dkogan/gnuplotlib and https://github.com/jtambasco/gnuplotpy. Chose `gnuplotlib`. I don't see why we could not install both.
- Numeric: raise SystemExit, "Python 2.0 or later required to build Numeric."
  - This was only support in Python 2.1
  - The replacement is numpy, which is already being installed.
- pychecker: SyntaxError: Missing parentheses in call to 'print'. Did you mean print("note: install_scripts can only be invoked by install")?
  - Since this is a development tools, I removed the source and this package, as is no longer supported.
  - Updated pylint to latest version 2.1.1
- numarray: SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Wrote config.h")?
  - numarray since 2004 is no longer developed. Its webpage encourages to use NumPy
  - Check https://www.scipy.org/scipylib/faq.html#id6 for more information.
  - We will install NumPy.
- pyephem: This project is for Python2. Replace by latest version of ephem.
- pysqlite: Deprecated in Python3 in favor of the included `sqlite3` module
- python-ldap not compatible with python3, updated to version 3.1.0
- pythoscope out of development. Since this is a development tool, we will provide instead test-generator.
- scipy updated to 1.1.0
- snakefood: version 1.4 claims its compatible, but the code differs from what it is pip, from the repository. Also, this is used by TELCAL.
  - Currently downloading the latest version from a cloned git repository. Not the nicest solution.
- suds: Does not work with python3. Installing instead suds-py3 (1.3.3.0)
- Twisted updated to 18.7.0
- astroid version 1.3.6 incompatible with pylint. Updated to 2.0.4.
- lxml updated to 4.2.4, as c interfaces are outdated.
- pyOpenSSL updated to 18.0.0
- Mock was removed in favor of included mock library. mock.path_object is not mock.patch.object. File object are now _io.IOStream, so you must use that for the class. Also, remember to decode from utf8, so you get the same output from IOStreams.

buildOmniORB

- Uploaded version 4.2.2 of omniORB-4.2.2.tar.bz2 and omniORBpy-4.2.2.tar.bz2
- Updated build script to use this versions
- TODO: There is this line, in which version dependant libraries are copied. I just put python3 there, but not sure if this will match the expected names.
  `mv $OMNI_ROOT/lib/python3.* $OMNI_ROOT/lib/python`

buildSwig

- TODO: Check if it works correctly for Python3 calls to C.
- Swig is currently in 3.0.12 version, which is the latest.

Python executable

- Any ACS module failes to compile, as several scripts try to find the `python` executable. Python 3 installs an exectuable that is called `python3`.
- We need to decide if we set in the path python as python3, or we change every makefile and script that calls python, as to use python3 instead.
  - I introduced a simbolic link in /alma/ACS-XXX/Python/bin/python. Several scripts try to identify python just by calling it, assuming it will be present in the path. The system most likely will include python 2.7 as `python`, according to base OS specification. It was decided in a first pass, that this will be the first approach used.

LGPL

acsBUILD

- ACS/LGPL/acsBUILD/config/.acs/.bash_profile.acs Has two small python scripts embedded.

Tools

extpy

- Removed pyxb and pmw, and moved them to acs.req
  - pyxb is used by xmlpybind.
- PyXML will be a complicated port. PyXML stopped at the version we currently have installed 8.0.4. Python3 includes its own libraries for xml parsing. It supports dom and sax.
  - We will remove PyXML and port to python3 libraries.
  - TODO: Do extensive testing on xml binding and processing in ACS with python.
  - TODO: PyXB is a library for XSD schema binding. Will have to check if python3 xml and pyxb are still compatible. Or if they were ever used in conjunction.
  - TODO: PyXB had a patch for an erroneous data type.
  - Element list does not support anymore "content". It needs to be replaced by "orderedContent".

```
== XSD Compiling with pyxbgen (Python): ACSError
Deprecated complexTypeDefinition method "content" invoked
Please use "orderedContent"
  File "/alma/ACS-2018JUN/ACSSW/bin/generateXsdPythonBinding", line 172, in <module>
    sys.exit(main(sys.argv[1:]))
  File "/alma/ACS-2018JUN/ACSSW/bin/generateXsdPythonBinding", line 150, in main
    flist,nslist = find_schema_files(ebs, xml.name)
  File "/alma/ACS-2018JUN/ACSSW/bin/generateXsdPythonBinding", line 55, in find_schema_files
    for elem in ebs.content():
```

- acs_python.py is a copy from a file inside the omniORBpy tarball. This file is the entry point for python .py files generation from an idl.
  - Comparing to the python2 version included in the 4.2.2 omniORBpy tarball, I can say that they are almost equal, exceptions are:

        - Some minor syntax changes
        - Missing package imports detection
        - Print out of imports in a different manner.
    - extidl makes use of this file
    - I took the python3 version of the file from omniORBpy-4.2.2, added the extra logic, and made the changes for python3 with 2to3.

Kit

acs

- Applied 2to3 to the whole src
- Checked indentation with pylint, and corrected several issues.
- acsMakefileDefinitions.mk is in charge of finding out the pyc files needed to copy when doing an `make install`. pyc files since python3.2 are mandatorily store in a __pycache__ folder, and the syntax of the file name has changed.
    - Added extra paths for modules and scripts and packages.
    - Corrected rules for clean target, to be able to delete the proper __pycache__/XYZ.cpython36.pyc files.
- `acsGetSpecificJars.py`: This is a python script that prepares the whole path to a list of jar files. This script failed silently in the make command, even with MAKE_VERBOSE=1. Corrected.

acsutilPy

- acsInstanceLock.py 2to3'ed
- ACSDirectory.py 2to3'ed
- ACSImport.py 2to3'ed
- AcsInstanceLockHelper.py 2to3'ed
- ACSPorts.py 2to3'ed
- FindFile.py 2to3'ed

ALL

Module	2to3	pylint	compiles /installs	Tests passes	Comments
ExtProd					
Boost	NOT DONE	NOT DONE	OK		Calling Boost's bootstrap with specific python version.
Python	OK	NOT DONE	OK		Updated to 3.6.6, added python to python3 link in the Makefile, added symbolic link to include, so boost can pick it up
PyModules	OK	NOT DONE	OK		Several libraries removed/updated/added
OmniORB	OK	NOT DONE	OK		Updated to 4.2.2. PYTHONPATH for omni ORB now includes python version.
Tcltk	OK	NOT DONE	OK		Added seqSh and seqWish links in $TCLTK_ROOT/bin in the Makefile
Kit					
acs	OK	OK	OK	OK	Modified rules in acsMakefile for ***pycache***. Severals scripts ported.
acsutilpy	OK	OK	OK	OK	Corrected the tests
Tools					
extpy	NOT DONE	NOT DONE	NOT DONE	OK	All modules moves to Tools/PyModules
xsddoc	NOT DONE	NOT DONE	NOT DONE	OK	
CommonSoftware					
xmlpybind	OK	OK	OK	OK	
acsstartup	OK	OK	OK	OK	subprocess returns bytes as stdout/err. https://stackoverflow.com/questions/606191/convert-bytes-to-a-string#606199 `mock` is included since python 3.3. Removing from dependencies.

loggingidl	PENDING	PENDING	OK	OK	
acserr	OK	OK	OK	OK	
acserrTypes	OK	OK	OK	OK	
baciidl	PENDING	PENDING	OK	OK	
loggingts	OK	OK	OK	OK	config had several code generator used only in testing. Python code generator needed porting to python3.
cdb	PENDING	PENDING	OK	OK	
cdb_rdb	OK	OK	OK	OK	Test case had a python script.
recovery	OK	OK	OK	OK	Test script ported to python3 and fixed indentation.
acstime	NOT DONE	NOT DONE	OK	OK	Had a swig generated python and c++ interface. Forced its regeneration.
acsnc	OK	PENDING	OK	NOT DONE	Test cases had python scripts. Ported to python3. Test cases failing to the same error as before.
acsdaemon	PENDING	PENDING	PENDING	PENDING	
acstestcompcpp	PENDING	PENDING	PENDING	PENDING	
acspycommon	OK	OK	OK	OK	XYZSeq has an increment method that was implemented through lambda. TEST.
acsalarmpy	OK	OK	OK	OK	
acspy	OK	OK	OK	OK	Indentation issues in python code.
nsStatisticsService	PENDING	PENDING	PENDING	PENDING	
jbaci	PENDING	PENDING	PENDING	PENDING	
monitoring	PENDING	PENDING	PENDING	PENDING	
acssamp	PENDING	PENDING	PENDING	PENDING	
acspyexmpl	OK	OK	OK	OK	
nctest	PENDING	PENDING	PENDING	PENDING	
acssim	OK	OK	OK	OK	TODO: FloatSeq should by in acspy
bulkDataNT	PENDING	PENDING	PENDING	PENDING	
containerTests	OK	OK	OK	OK	
* contLogTest	OK	OK	OK	OK	
* conNcTests	OK	OK	OK	OK	
acscourse	PENDING	PENDING	PENDING	PENDING	
ACSLaser (demo)	OK	OK	OK	OK	
acsncdds	PENDING	PENDING	PENDING	PENDING	

List of files that could contain python code (and are not .py files)

```
acserr/ws/config/AES2Py.xslt:<xsl:text>#!/usr/bin/env python
acserr/ws/test/pythonTest.sh:python test_AES2Py.py
acserr/ws/test/Makefile:SCRIPTS_L       = acserrTestRun.sh startAcserrTest.sh testJUnitACSERRnoRuntime pythonTest
acsGUIs/alarmPanel/test/Makefile:# jagonzal: These are shell scripts not python scripts
ACSLaser/baciPropsTest/test/Makefile:# jagonzal: These are python scripts not shell scripts
ACSLaser/alarmTests/test/Makefile:# jagonzal: This is a shell script not a python script
ACSLaser/demo/test/Makefile:# jagonzal: These are shell scripts not python scripts
acsncdds/test/testCDB:#!/usr/bin/env python
acsncidl/ws/test/acsncidlPy:python -c 'from acsnc import EventDescription'
acspy/bin/acspyInteractiveContainer:exec python -i $PYBINARY $@ -interactive
acspy/bin/ACSStartContainerPy:#!/usr/bin/env python
acspy/bin/ACSPyConsole:#!/usr/bin/env python
acspy/bin/ACSPyConsole:Starts an ACS interactive python console.
acspy/bin/acsLoggingMonitor:#!/usr/bin/env python
acspy/src/acspyInteractiveContainer:exec python -i $PYBINARY $@ -interactive
acspy/src/acsLoggingMonitor:#!/usr/bin/env python
acspycommon/test/acspyTestLoggingStatistics.sh:LOCATION=TST-STE python ../bin/acspyTestLoggingStatistics 2>&1;
acspycommon/test/acspyTestLoggingStatistics.sh:python ../bin/acspyTestLoggingStatistics 2>&1;
acssim/config/CDB/schemas/SimulatedComponent.xsd:                                <xs:element name="pythonImports"
type="imports" minOccurs="0"/>
acssim/src/recordEvents:#!/usr/bin/env python
acsstartup/bin/acsstartupCreateChannel:#! /usr/bin/env python
acsstartup/bin/acsstartupAcsPorts:    python -c "import socket; import os;  print str(socket.gethostbyname(os.
environ['ACS_HOST']))"
acsstartup/bin/acsstartupAcsPorts:        python -c "import socket; print(str(socket.gethostbyname(socket.
getfqdn())))"
acsstartup/bin/acsstartupAcsPorts:        python -c "import socket; print(str(socket.gethostbyname(socket.
getfqdn().split('.')[0])))"
acsstartup/bin/acsdataClean:#!/usr/bin/env python
acsstartup/bin/killACS:#!/usr/bin/env python
acsstartup/bin/acsstartupRemovePID:#!/usr/bin/env python
acsstartup/bin/acsNotifysStatus:#!/usr/bin/env python
acsstartup/bin/acsstartupNotifyPort:#!/usr/bin/env python
acsstartup/bin/acsstartupContainerPort:#!/usr/bin/env python
acsstartup/bin/acsContainersStatus:#!/usr/bin/env python
acsstartup/src/acsstartupAcsPorts:    python -c "import socket; import os;  print(str(socket.gethostbyname(os.
environ['ACS_HOST'])))"
acsstartup/src/acsstartupAcsPorts:        python -c "import socket; print(str(socket.gethostbyname(socket.
getfqdn())))"
acsstartup/src/acsstartupAcsPorts:        python -c "import socket; print(str(socket.gethostbyname(socket.
getfqdn().split('.')[0])))"
acstime/ws/src/acstimeSWIG_wrap.cpp:/* http://www.python.org/dev/peps/pep-0353/#conversion-guidelines */
acstime/ws/src/acstimeSWIG_wrap.cpp:/* The python void return value */
acstime/ws/src/acstimeSWIG_wrap.cpp:      printf("swig/python detected a memory leak of type '%s', no destructor
found.\n", name);
acstime/ws/src/acstimeSWIG_wrap.cpp:   is copied out of Python/modsupport.c in python version 2.3.4 */
acstime/ws/src/acstimeSWIG_wrap.cpp:/* The python cached type query */
acstime/ws/src/acstimeSWIG_wrap.cpp:#  error "This python version requires swig to be run with the '-classic'
option"
acstime/ws/src/acstimeSWIG_wrap.cpp:       In python the user should not be able to modify the inner
acstime/ws/src/acstimeSWIG_wrap.cpp:       SWIG_PYTHON_SAFE_CSTRINGS, a new/copy of the python string
acstime/ws/src/Makefile:_acstimeSWIG_LIBS      = baci ACSTimeError maciErrType python2.7
acstime/ws/src/Makefile:        @$(RM) $(INSTALL_ROOT)/lib/python/site-packages/_acstimeSWIG.$(SHLIB_EXT)
acstime/ws/src/Makefile:        @cp ../lib/lib_acstimeSWIG.$(SHLIB_EXT) $(INSTALL_ROOT)/lib/python/site-packages
/_acstimeSWIG.$(SHLIB_EXT)
acstime/ws/src/Makefile:        @swig -c++ -python ../config/acstimeSWIG.i
acsutil/ws/bin/acsutilProfiler: elapsed=`python -c "print $stop - $start"`
acsutil/ws/bin/acsutilProfiler: totalTime=`python -c "print $elapsed + $totalTime"`
acsutil/ws/bin/acsutilProfiler: minDuration="`python -c "print min($minDuration,$elapsed)"`"
acsutil/ws/bin/acsutilProfiler: maxDuration="`python -c "print max($maxDuration,$elapsed)"`"
acsutil/ws/bin/acsutilProfiler:averageTime=`python -c "print $totalTime / $CL_RUNS"`
acsutil/ws/src/python:  PYTHONSHELL="exec $PYTHON_ROOT/bin/python"
acsutil/ws/src/acsutilProfiler: elapsed=`python -c "print $stop - $start"`
acsutil/ws/src/acsutilProfiler: totalTime=`python -c "print $elapsed + $totalTime"`
acsutil/ws/src/acsutilProfiler: minDuration="`python -c "print min($minDuration,$elapsed)"`"
acsutil/ws/src/acsutilProfiler: maxDuration="`python -c "print max($maxDuration,$elapsed)"`"
acsutil/ws/src/acsutilProfiler:averageTime=`python -c "print $totalTime / $CL_RUNS"`
baciidl/ws/test/baciidlPy:python -c 'from maci import Container'
cdbChecker/test/testdata/defaultCDB/CDB/schemas/SimulatedComponent.xsd:                        <xs:element name="
pythonImports" type="imports" minOccurs="0"/>
loggingts/ws/config/LTS2Py.xslt:<xsl:text>#! /usr/bin/env python
maciidl/ws/test/maciidlPy:python -c 'from maci import Container'
xmlpybind/bin/generateXsdPythonBinding:#! /usr/bin/env python
```

acserr

- compileall.py from omniORB fails to find ACSERR module/package.
- This fails because xmlpybind EntitybuilderSettings.py is failing.
- Fixing xmlpybind and acsstartup revealed that there is a python module in the source of this code.
- ACSErrorChecker was converted using 2to3

- TODO: ErrorDefinition.py had several indentation issues, and what I consider, coding errors also. Will need a review on this code.
- There is an XSLT used to transform xml ACSERR definitions into python code. This was updated to python3.

xmlpybind

- This modules takes an XSD from xmljbind, EntitybuildSetting.xsd, and code generate using pyxb a python class that is able to contain namespace versus schemas associations, from different (XML?) files.
- The class produced from the older version of pyxb to the new one is quite different. The XSD is just one complex type, composed of a sequence of two complex type.
- The error here, is that the class `EntitybuilderSettings.py` is used by `generateXsdPythonBinding.py`, in the same ACS module. It iterates over the returns content, which is an attribute no longer available. Replacing it by orderedContent did no good.
- TODO: Low priority. We had to go back to the ebs.content() method to navigate the contents of the parsed xml, as the orderedContent() returns a different kind of objects.

acsstartup

- acsContainersStatus.py 2to3'ed
- acsdataClean.py 2to3'ed
- acsNotifysStatus.py 2to3'ed
- acsstartupContainerPort.py 2to3'ed
- acsstartupCreateChannel.py 2to3'ed
- acsstartupNotifyPort.py 2to3'ed
- acsstartupRemovePID.py 2to3'ed
- killACS.py 2to3'ed
- IMPORTANT: Python3 has a unified integer datatype which behaves like a long from python2, it is called `int`.
- acsstartupAcsPorts has inline python code.

Notes for migration from Python2 to Python3.

- `2to3` and `pylint` will be your friends
- `2to3` does an excellent job, but there are still some things it does not handle:
  - opening a file. file('bla.bla','r') is now open('bla.bla','r')
  - string.split() method now is part of the string object.
- Recommendation: run `2to3 -w -n src -o src.new`. This will take everything in src, convert and save it to src.new.
  - Then use `meld` to compare both directories, and copy what you determine is correct.
- `pylint` can be used to detect and manually correct indentation issues. Python3 is more strict with indentation. Several parts of the code needed to be clearly understood, as code blocks in python without indentation do not necessaryly make sense on its own.
- pyxml was removed, and we will use now that xml library that comes with Python3.
- pyxb was updated, and now generated a depracted warning. This appears when the acsMakefile makes use of XSD binding.
- Python3 generates pyc files in a difference location, following: __pycache__/XYZ.cpython36.pyc. This has been changed also in the acsMakefiles. There were though, some ACS modules that deleted them on the clean target manually.
- To launch now a process, the python module is subprocess. This is picked up by 2to3, but the output is in Bytes, not a string. This needs to be decoded into UTF8. See https://stackoverflow.com/questions/606191/convert-bytes-to-a-string#606199
- Python3 uses as a Long datatype the int keyword. This can be confusing, given that in Python2, both int and long meant different things. Long is now the keyword `int`.
- Several bash scripts have inline python code. This seems like a common practice. Keep an eye for those.