The CDB Simulated Components are a quick way to make a component available for development and testing. They could be used to prototype basic functionality and to mock-up components in order to prepare module tests which need to interact with other components.

Create a module for the tests that are going to be performed:

```
> getTemplateForDirectory MODROOT_WS sim_comp
> cd sim_comp
```

Put the IDL file:

sim_comp/idl/MountExample.idl

```
#ifndef _MOUNT_EXAMPLE_IDL_
#define _MOUNT_EXAMPLE_IDL_

#include <acscomponent.idl>

#pragma prefix "alma"

module workshop
{
    interface MountExample : ACS::ACSComponent {
        void objfix (in double az, in double elev);
    };
};

#endif
```

Configure the Manager in the CDB:

sim_comp/test/CDB/MACI/Managers/Manager/Manager.xml

```
<Manager xmlns:cdb="urn:schemas-cosylab-com:CDB:1.0" xmlns="urn:schemas-cosylab-com:Manager:1.0" xmlns:log="urn:
schemas-cosylab-com:LoggingConfig:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Timeout="50.0"
ContainerPingInterval="20.0">
    <Startup> </Startup>
    <ServiceComponents>
        <cdb:e string="Log"/>
        <cdb:e string="LogFactory"/>
        <cdb:e string="NotifyEventChannelFactory"/>
        <cdb:e string="MC_NotifyEventChannelFactory" />
        <cdb:e string="LoggingNotifyEventChannelFactory" />
        <cdb:e string="MC_LoggingNotifyEventChannelFactory" />
        <cdb:e string="LoggingChannel@LOGGING.channels"/>
        <cdb:e string="ArchivingChannel@ARCHIVING.channels"/>
        <cdb:e string="AlarmNotifyEventChannelFactory" />
        <cdb:e string="MC_AlarmNotifyEventChannelFactory" />
        <cdb:e string="InterfaceRepository"/>
        <cdb:e string="AlarmChannel" />
        <cdb:e string="CDB"/>
        <cdb:e string="ACSLogSvc"/>
        <cdb:e string="AcsAlarmService"/>
    </ServiceComponents>
    <LoggingConfig>
        <log:e Name="jacorb@Manager" minLogLevel="5" minLogLevelLocal="5"/>
    </LoggingConfig>
</Manager>
```

Configure the Component in the CDB:

sim_comp/test/CDB/MACI/Components/Components.xml

```
<Components xmlns="urn:schemas-cosylab-com:Components:1.0" xmlns:cdb="urn:schemas-cosylab-com:CDB:1.0" xmlns:
baci="urn:schemas-cosylab-com:BACI:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<e Name="MOUNT" Code="Acssim.Servants.Simulator"
   Type="IDL:alma/ACS_COURSE/MountExample:1.0"
   Container="pyContainer"
   ImplLang="py"/>
</Components>
```

Prepare the Makefile:

```
> cp src/Makefile test/Makefile
```

Add the following to IDL entry:

sim_comp/test/Makefile

```
...
#
# IDL Files and flags
#
IDL_FILES = MountExample
TAO_IDLFLAGS =
USER_IDL =
MountExampleStubs_LIBS = acscomponentStubs
...
```

Generate code stubs and skeletons:

```
> cd test
> make all
```

The simulation code needs to be put in the CDB:

sim_comp/test/CDB/alma/simulated/MOUNT/MOUNT.xml

```
<SimulatedComponent xmlns="urn:schemas-cosylab-com:SimulatedComponent:1.0" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance">
<pythonImports>
from acstime import Duration
</pythonImports>
<initialize>
logger=parameters[0].getLogger()
logger.logInfo("Component initialized!")
</initialize>
<cleanUp>
print('Simulated component cleaning up')
</cleanUp>
<_corbaMethod Name="objfix" Timeout="0">
print("\nobjfix called with params ", parameters, "\n")
</_corbaMethod>
</SimulatedComponent>
```

A simple client could be prepared:

sim_comp/test/client.py

```
from Acspy.Clients.SimpleClient import PySimpleClient
client = PySimpleClient()
m = client.getComponent("MOUNT")
m.objfix(3,6)
```

Start ACS:

Console 1

```
> cd sim_comp
> export ACS_CDB=$PWD/test
> export IDL_PATH="-I$PWD/idl $IDL_PATH"
> acsStart
```

On a different console start pyContainer:

Console 2

```
> cd sim_comp
> export PYTHONPATH=$PWD/lib/python/site-packages:$PYTHONPATH
> acsStartContainer -py pyContainer
```

Execute objexp or the client in a third console:

Console 3

```
> export PYTHONPATH=$PWD/lib/python/site-packages:$PYTHONPATH
> python test/client.py
```

The expected behavior is that no error is shown in "Console 3" and "Console 2 shold have the following output:

Console 2 Output

```
2020-07-25T23:45:52.198 INFO [acsStartContainer] Running the container with these arguments:
'ACSStartContainerPy pyContainer -ORBendPoint giop:tcp:127.0.1.1:4000 -m corbaloc::127.0.1.1:3000/Manager'
ContainerStatusMsg: Startup begins
2020-07-25T23:45:53.027 AlarmSystemInterfaceFactory init - Using ACS alarm system
ContainerStatusMsg: ORB initialization begins
ContainerStatusMsg: ORB initialization ends
ContainerStatusMsg: Manager access initialization begins
ContainerStatusMsg: Manager access initialization ends
ContainerStatusMsg: Automatic component loading begins
2020-07-25T23:45:53.077 pyContainer taggedmessage - Info message from the manager: Startup statistics: 0
components queued to be activated.
2020-07-25T23:45:53.079 pyContainer taggedmessage - Info message from the manager: Startup statistics: 0 of 0
components activated.
ContainerStatusMsg: Automatic component loading ends
ContainerStatusMsg: Ready
2020-07-25T23:45:53.168 pyContainer getCDBInfo - No container information found in the CDB
2020-07-25T23:45:53.183 pyContainer __init__ - Container pyContainer waiting for requests
2020-07-25T23:45:53.187 pyContainer __init__ - Container pyContainer used libraries from /home/almamgr/mod
/src/../lib/python/site-packages/:/alma/ACS-2020AUG/ACSSW/lib/python/site-packages:
ContainerStatusMsg: Startup ends
2020-07-25T23:45:54.869 pyContainer activate_component_async - Starting async activation of MOUNT(type IDL:alma
/workshop/MountExample:1.0)
2020-07-25T23:45:54.871 pyContainer run - Activating MOUNT (type IDL:alma/workshop/MountExample:1.0)
2020-07-25T23:45:54.872 pyContainer aboutToActivate - The container is asking the clearance to activate MOUNT
2020-07-25T23:45:54.873 pyContainer _aboutToProcess - The container is not dealing with MOUNT: the operation
can go ahead
2020-07-25T23:45:54.877 pyContainer aboutToActivate - The container has been cleared to activate MOUNT
2020-07-25T23:45:55.075 loggingts -- Successfully loaded component code. [ logName=LOG_CompAct_Loading_OK
TimeMillis=194 CompName=MOUNT ]
2020-07-25T23:45:55.130 loggingts -- Successfully instantiated component. [ logName=LOG_CompAct_Instance_OK
TimeMillis=47 CompName=MOUNT ]
2020-07-25T23:45:55.442 MOUNT stringFunction - Component initialized!
2020-07-25T23:45:55.444 Servants.Executor _execute - initialize return value looks like:None of type:<class
'NoneType'>
2020-07-25T23:45:55.448 EventDispatcher (MOUNT) setupEventDispatching - Setting up event dispatching.
2020-07-25T23:45:55.459 EventDispatcher (MOUNT) handleFrequencies - No event frequencies defined.
2020-07-25T23:45:55.463 EventDispatcher (MOUNT) handleResponses - No event frequencies defined.
2020-07-25T23:45:55.508 loggingts -- Successfully activated component with Corba. [
logName=LOG_CompAct_Corba_OK TimeMillis=0 CompName=MOUNT ]
2020-07-25T23:45:55.510 loggingts -- Successfully initialized component. [ logName=LOG_CompAct_Init_OK
TimeMillis=0 CompName=MOUNT ]
2020-07-25T23:45:55.560 pyContainer activate_component - Activated component: MOUNT
2020-07-25T23:45:55.571 pyContainer activate_component - Component 'MOUNT' has KeepAliveTime '0'.
2020-07-25T23:45:55.576 pyContainer activated - The container terminated the activation of MOUNT: awakening
waiting threads
2020-07-25T23:45:55.579 pyContainer _processTerminated - The container terminated dealing with MOUNT
2020-07-25T23:45:55.582 pyContainer run - Calling maci::CBComponentInfo::done with descOut.id_tag = 1 for
'MOUNT'
2020-07-25T23:45:55.601 pyContainer run - Call to maci::CBComponentInfo::done with descOut.id_tag = 1 for
'MOUNT' completed

objfix called with params  [3.0, 6.0, <Acssim.Servants.Simulator.Simulator_IDL:alma/workshop/MountExample:1.0
object at 0x7f0608194d30>]

2020-07-25T23:45:55.610 Servants.Executor _execute - objfix return value looks like:None of type:<class
'NoneType'>
2020-07-25T23:45:55.689 pyContainer aboutToDeactivate - The container is asking the clearance to deactivate
MOUNT
2020-07-25T23:45:55.693 pyContainer _aboutToProcess - The container is not dealing with MOUNT: the operation
can go ahead
2020-07-25T23:45:55.696 pyContainer aboutToDeactivate - The container has been cleared to activate MOUNT
2020-07-25T23:45:55.736 pyContainer deactivate_component - Deactivating component: MOUNT
Simulated component cleaning up
2020-07-25T23:45:55.741 Servants.Executor _execute - cleanUp return value looks like:None of type:<class
'NoneType'>
2020-07-25T23:45:55.747 pyContainer deactivated - The container terminated the deactivation of MOUNT: awakening
waiting threads
2020-07-25T23:45:55.749 pyContainer _processTerminated - The container terminated dealing with MOUNT
```