

- [Introduction](#)
- [Presentation](#)
- [Hands-On Exercise](#)
 - [Component](#)
 - [Configuration](#)
 - [Request \(Python\)](#)
 - [Default Component](#)
 - [Configuration](#)
 - [Request \(Python\)](#)
 - [Non Sticky Component](#)
 - [Configuration](#)
 - [Request \(Python\)](#)
 - [Collocated Component](#)
 - [Configuration](#)
 - [Request \(Python\)](#)
 - [Dynamic Component](#)
 - [Configuration](#)
 - [Request \(Python\)](#)
 - [Auto Start Component](#)
 - [Configuration](#)
 - [Request \(Python\)](#)
 - [Immortal Component](#)
 - [Configuration](#)
 - [Request \(Python\)](#)
 - [Delayed Deactivation Component](#)
 - [Configuration](#)
 - [Request \(Python\)](#)
- [Discussion](#)

There are several types of components in the ALMA Common Software and they're useful in different contexts. The list of component types is the following:

- Component
- Default Component
- Non Sticky Component
- Collocated Component
- Dynamic Component
- AutoStart Component
- Immortal Component
- Delayed Deactivation Component

Each of the components type differentiates of each other in one or more of the following areas:

- Configuration
- Request Procedure
- Lifecycle Behavior
- Scope
 - Describe Component Types
 - Particularities on the 3 areas for each component type
- Duration: 10 minutes

Component

Configuration

Components.xml

```
<e Name="CLOCKTEST"
  Code="acsclock"
  Type="IDL:alma/acstime/Clock:1.0"
  Container="bilboContainer"
  ImplLang="cpp"
  KeepAliveTime="0"
  Autostart="false"
  Default="false"
>
</e>
```

Request (Python)

```

from Acsapy.Clients.SimpleClient import PySimpleClient
client = PySimpleClient()
compName = 'CLOCKTEST'

compInfo = client.availableComponents(compName)
if len(compInfo) != 1:
    print("Problem with the component: there should be one definition.")
compInfo = compInfo[0]
if not compInfo.reference is None:
    print("Problem with the component: it is already active.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone already has a reference to it.")

clock = client.getComponent(compName)
compInfo = client.availableComponents(compName)[0]
if compInfo.reference is None:
    print("Problem with the component: it should have been activated.")
if len(compInfo.clients) != 1:
    print("Problem with the component: it has not been activated or more than one component/client have a
reference to it.")

client.releaseComponent(clock.name)
compInfo = client.availableComponents(compName)[0]
if not compInfo.reference is None:
    print("Problem with the component: it should have been deactivated.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone still has a reference to it.")

```

Default Component

Configuration

Components.xml

```

<e Name="CLOCKTEST"
  Code="acsclock"
  Type="IDL:alma/acstime/Clock:1.0"
  Container="bilboContainer"
  ImplLang="cpp"
  KeepAliveTime="0"
  Autostart="false"
  Default="true"
>
</e>

```

Request (Python)

```

from Acsapy.Clients.SimpleClient import PySimpleClient
client = PySimpleClient()
compName = 'CLOCKTEST'

compInfo = client.availableComponents(compName)
if len(compInfo) != 1:
    print("Problem with the component: there should be one definition.")
compInfo = compInfo[0]
if not compInfo.reference is None:
    print("Problem with the component: it is already active.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone already has a reference to it.")

clock = client.getDefaultComponent('IDL:alma/acstime/Clock:1.0')
compInfo = client.availableComponents(compName)[0]
if compInfo.reference is None:
    print("Problem with the component: it should have been activated.")
if len(compInfo.clients) != 1:
    print("Problem with the component: it has not been activated or more than one component/client have a
reference to it.")

client.releaseComponent(clock.name)
compInfo = client.availableComponents(compName)[0]
if not compInfo.reference is None:
    print("Problem with the component: it should have been deactivated.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone still has a reference to it.")

```

Non Sticky Component

Configuration

Components.xml

```

<e Name="CLOCKTEST"
  Code="acsclock"
  Type="IDL:alma/acstime/Clock:1.0"
  Container="bilboContainer"
  ImplLang="cpp"
  KeepAliveTime="0"
  Autostart="false"
  Default="false"
>
</e>

```

Request (Python)

```

from Acsapy.Clients.SimpleClient import PySimpleClient
client = PySimpleClient()
compName = 'CLOCKTEST'

compInfo = client.availableComponents(compName)
if len(compInfo) != 1:
    print("Problem with the component: there should be one definition.")
compInfo = compInfo[0]
if not compInfo.reference is None:
    print("Problem with the component: it is already active.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone already has a reference to it.")

tmp = client.getComponent(compName)
compInfo = client.availableComponents(compName)[0]
if compInfo.reference is None:
    print("Problem with the component: it should have been activated.")
if len(compInfo.clients) != 1:
    print("Problem with the component: it has not been activated or more than one component/client have a
reference to it.")

clock = client.getComponentNonSticky(compName)
compInfo = client.availableComponents(compName)[0]
if len(compInfo.clients) != 1:
    print("Problem with the component: it has not been activated or more than one component/client have a
reference to it.")

client.releaseComponent(tmp.name)
compInfo = client.availableComponents(compName)[0]
if not compInfo.reference is None:
    print("Problem with the component: it should have been deactivated.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone still has a reference to it.")

```

Collocated Component

Configuration

Components.xml

```

<e Name="CLOCKTEST"
  Code="acsclock"
  Type="IDL:alma/acstime/Clock:1.0"
  Container="bilboContainer"
  ImplLang="cpp"
  KeepAliveTime="0"
  Autostart="false"
  Default="false"
>
</e>

```

Request (Python)

```

from Acsapy.Clients.SimpleClient import PySimpleClient
import maci

client = PySimpleClient()
compSpec = maci.ComponentSpec("CLOCKTEST", "", "", "")
comp = client.getCollocatedComp(compSpec, False, "CLOCKTEST")

```

Dynamic Component

Configuration

Components.xml

```
<e Name="*"
  Code="acsclock"
  Type="IDL:alma/acstime/Clock:1.0"
  Container="*"
  ImplLang="cpp"
  KeepAliveTime="0"
  Autostart="false"
  Default="false"
>
</e>
```

Request (Python)

```
from Acspy.Clients.SimpleClient import PySimpleClient
client = PySimpleClient()
compName = 'CLOCKTEST'
compType = 'IDL:alma/acstime/Clock:1.0'
container = bilboContainer

compInfo = client.availableComponents(compName)
if len(compInfo) != 0:
    print("Problem with the component: there shouldn't be any definition.")

clock = client.getDynamicComponent(compName, '*', compType, container)
compInfo = client.availableComponents(compName)
if len(compInfo) != 1:
    print("Problem with the component: there should be one definition.")
compInfo = compInfo[0]
if compInfo.reference is None:
    print("Problem with the component: it should have been activated.")
if len(compInfo.clients) != 1:
    print("Problem with the component: it has not been activated or more than one component/client have a reference to it.")

client.releaseComponent(clock.name)
compInfo = client.availableComponents(compName)[0]
if not compInfo.reference is None:
    print("Problem with the component: it should have been deactivated.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone still has a reference to it.")
```

Auto Start Component

Configuration

Components.xml

```
<e Name="CLOCKTEST"
  Code="acsclock"
  Type="IDL:alma/acstime/Clock:1.0"
  Container="bilboContainer"
  ImplLang="cpp"
  KeepAliveTime="0"
  Autostart="true"
  Default="false"
>
</e>
```

Request (Python)

```
from Acspy.Clients.SimpleClient import PySimpleClient
client = PySimpleClient()
compName = 'CLOCKTEST'

compInfo = client.availableComponents(compname)
if len(compInfo) != 1:
    print("Problem with the component: there should be one definition.")
compInfo = compInfo[0]
if compInfo.reference is None:
    print("Problem with the component: It should be already active.")
if len(compInfo.clients) != 1:
    print("Problem with the component: There should be only one reference to it.")
if compInfo.clients[0] != 83886080:
    print("Problem with the component: The manager should have a reference to it.")

clock = client.getComponent(compName)
compInfo = client.availableComponents(compName)[0]
if len(compInfo.clients) != 2:
    print("Problem with the component: There should be 2 references; 1 from manager and the other from our client.")

client.releaseComponent(clock.name)
compInfo = c.availableComponents(compName)[0]
if compInfo.reference is None:
    print("Problem with the component: It shouldn't be deactivated.")
if len(compInfo.clients) != 1:
    print("Problem with the component: Only the Manager should have a reference to it.")
```

Immortal Component

Configuration

Components.xml

```
<e Name="CLOCKTEST"
  Code="acsclock"
  Type="IDL:alma/acstime/Clock:1.0"
  Container="bilboContainer"
  ImplLang="cpp"
  KeepAliveTime="-1"
  Autostart="false"
  Default="false"
>
</e>
```

Request (Python)

```
from Acspy.Clients.SimpleClient import PySimpleClient
client = PySimpleClient()
compName = 'CLOCKTEST'

compInfo = c.availableComponents(compName)
if len(compInfo) != 1:
    print("Problem with the component: there should be one definition.")
compInfo = compInfo[0]
if not compInfo.reference is None:
    print("Problem with the component: it is already active.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone already has a reference to it.")

clock = client.getComponent()
compInfo = c.availableComponents(compName)[0]
if compInfo.reference is None:
    print("Problem with the component: it should have been activated.")
if len(compInfo.clients) != 2:
    print("Problem with the component: There should be 2 references; 1 from manager and the other from our client.")

client.releaseComponent(clock.name)
compInfo = c.availableComponents(compName)[0]
if compInfo.reference is None:
    print("Problem with the component: It shouldn't be deactivated.")
if len(compInfo.clients) != 1:
    print("Problem with the component: Only the Manager should have a reference to it.")
```

Delayed Deactivation Component

Configuration

Components.xml

```
<e Name="CLOCKTEST"
  Code="acsclock"
  Type="IDL:alma/acstime/Clock:1.0"
  Container="bilboContainer"
  ImplLang="cpp"
  KeepAliveTime="10"
  Autostart="false"
  Default="false"
>
</e>
```

Request (Python)

```

import time
from Acspy.Clients.SimpleClient import PySimpleClient
client = PySimpleClient()
compName = 'CLOCKTEST'

compInfo = c.availableComponents(compName)
if len(compInfo) != 1:
    print("Problem with the component: there should be one definition.")
compInfo = compInfo[0]
if not compInfo.reference is None:
    print("Problem with the component: it is already active.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone already has a reference to it.")

clock = client.getComponent(compName)
compInfo = c.availableComponents(compName)[0]
if compInfo.reference is None:
    print("Problem with the component: it should have been activated.")
if len(compInfo.clients) != 2:
    print("Problem with the component: There should be 2 references; 1 from manager and the other from our client.")

client.releaseComponent(clock.name)
compInfo = c.availableComponents(compName)[0]
if compInfo.reference is None:
    print("Problem with the component: It shouldn't be deactivated.")
if len(compInfo.clients) != 1:
    print("Problem with the component: Only the Manager should have a reference to it.")

time.sleep(20)
compInfo = c.availableComponents(compName)[0]
if not compInfo.reference is None:
    print("Problem with the component: it should have been deactivated.")
if len(compInfo.clients) > 0:
    print("Problem with the component: someone still has a reference to it.")

```

- New Component Types Ideas?
- Improvements