

- 

[Information on ACS\\_INSTANCE and ports](#)

Port Allocation

- [Daemon ports](#)
- [ACS services ports](#)
- [ACS container ports](#)
- [Client ports](#)
- [Background info on using ports in ACS](#)
- The environment variable ACS\_INSTANCE was introduced to allow developers / testers to run several independent instances of ACS and related software on one machine, without getting side effects other than reduced resources.
- In operational environments this variable should not be used, or should be set to 0.
  - instance 0 is the default set in .acs/.bash\_profile
  - instance 0 is privileged to other instance numbers because it allows running more containers per machine, see below.
- See also [FAQAcInstance](#)
- Allowed value for ACS\_INSTANCE are 0-9
- All ACS services (not daemons!) calculate their port number using ACS\_INSTANCE as an offset.

ACS uses certain port ranges only, to avoid conflicts among ACS processes (especially when running multiple ACS\_INSTANCES), but also to be more predictable for other software.

There are several port ranges reserved / used by ACS:

## Daemon ports

An ACS daemon is run as a single instance per computer, using a fixed port independently of \$ACS\_INSTANCE because it handles all acs instances. The same applies to the "Imp" child processes of the services daemon.

ACS reserves the port range 2900 - 2999 for daemon ports.

Currently we use:

Daemon	Fixed Port
Container Daemon	2970 (was 3013)
Services Daemon	2980 (was 3014)
Naming Service Imp	2981 (was 3015)
Notification Service Imp	2982 (was 3016)
CDB Imp	2983 (was 3017)
Manager Imp	2984 (was 3018)
acsLogSvc Imp	2985 (was 3019)
Log service Imp	2986 (was 3020)
IR Imp	2987 (was 3021)
Alarm service Imp	2988

## ACS services ports

ACS reserves the port range 3000 - 3999 for services ports.

The actual port number is calculated using the following formula:

$$\text{port} = 3000 + \text{ACS\_INSTANCE} * 100 + \text{port offset}$$

which means 50 possible port offsets for ACS services per ACS instance (the other 50 are for containers, see below). Currently we use

Service	Port Offset
Manager	0
Naming Service	1
Notify Service	2

Logging Service	3
Interface repository	4
(Logging) notification service	5
(BACI monitor archiving) notification service	6
(Alarm) notification service	7
ACS Log Service (acsLogSvc)	11
CDB (cdbjDAL) Service	12
Alarm Service	13
Additional Notify Service instances (see <a href="#">COMP-3432</a> )	20-49

*How about federated manager instances?*

## ACS container ports

ACS reserves the port range 4000 - 4999 for container ports.

However the allocation of container ports depends on the ACS instance, with instance 0 getting the large range 4000-4999, while containers from other ACS instances each have to fit in the range of 100 ports that they share with the ACS services, according to the same offset formula.

ACS Containers	ACS_INSTANCE 0: 4000 to 4999 (even numbers) ACS_INSTANCE > 0: port offset 50 to 72 (even numbers)
Java containers ports for JVMDI remote debugging	Allocated range is (odd numbers paired with container number) 50 to 74
Maci Supervisor clients	75 to 99 (see explanation below)

## Client ports

Maci supervisor clients share the port offset range 75 - 99 (above the container port range).

This may be not necessary any more. See comment "Ports for Maci Supervisors" below for the historical context.

Other client applications using C++ SimpleClient, Java ComponentClient etc pick ports freely, typically in a higher port range > 40.000.

## Background info on using ports in ACS

Utility APIs/classes are provided to access dynamic port numbers:

- C++ - acsutilPorts.h
- Java - alma.acs.util.ACSPorts
- Python - [AcsutilPy](#).ACSPorts
- Bash - acsstartupAcsPorts (source it)

APIs and utilities are available to calculate proper port numbers.

Notice that persistent port numbers in services and containers are essential to the implementation of persistent connections, i.e. connections that are still valid after a restart of the service/container/component. Therefore when re-starting ACS services or containers, their ORBs must use the same port as in the previous run. For containers, the scripts from module `acsstartup` protect and find these ports in `$ACSDATA/tmp/$ACS_INSTANCE/USED_CONTAINER_PORTS`, while for the services most of them have fixed ports anyway, and only for the additional notify services the daemons take care of protecting the ports.

Does not really belong here but may be interesting: The Alma Exec subsystem uses port 10162 as default for SNMP traps.