



Usage of Glue Schema v1.3 for WLCG Installed Capacity information

Editor: Flavia Donno

Date: 2/2/2009

Version: 1.8-6

Contributors/Authors:

Stephen Burke (GLUE/RAL), Greig Cowan (Storage Accounting/Edinburgh), Flavia Donno (WG Coordinator/CERN), Laurence Field (Glue/CERN) Jens Jensen ([RAL](#)), Michel Jouvin (DPM/GRIF), Miguel Marques Coelho Dos Santos (CASTOR/CERN), Luca Magnoni ([StoRM/CNAF](#)), Paul Millar (dCache/DESY), Jason Shih (CASTOR/Taiwan), Jeff Templon (NIKHEF), Steve Traylen (WN WG/CERN), Ron Trompert (dCache/SARA), Claudiu Schiaua (ALICE/NIPNE), Jan Van Eldik (CASTOR/CERN), Riccardo Zappi ([StoRM/CNAF](#)), Burt Holzman, Brian Bockelman (OSG)

Purpose of this document

The goal of this document is to detail how to use the Glue Schema version 1.3 in order to publish information to provide the WLCG management with a view of the total installed capacity and resource usage by the VOs at sites. This information can also be used by VO operations and management in order to monitor the VO usage of the resources:

1. Publishing information to provide the WLCG management with a view of the "total installed capacity" at a site. This includes resources that have not yet been configured/assigned to a VO or resources that are temporary down or offline. Measurements are not required to be 100% accurate or 100% reliable. Daily reports can be used by the cesga portal in order to come up with the monthly reports requested by the management.
2. Publishing information to provide the WLCG and VO management with a view of the resource assignment per VO at sites. This includes only resources that have been configured and explicitly assigned to a VO. An exception is made for shared resources. In this case it is accepted to provide only aggregated information for the common usage. Measurements are not required to be 100% accurate or 100% reliable. A 10% accuracy is acceptable. Daily reports can be used by the cesga

PUBLIC



February 2, 2009

- portal in order to come up with the monthly reports requested by the management.
3. Publishing information to allow VO operators to monitor the VO usage of the resources. This is specifically important for monitoring the usage of storage staging areas or to monitor the usage of disk only areas.
 4. Allowing clients to select resources based on their advertised characteristics and ensure that at least some agreed set of the advertised characteristics have a clear meaning. The selection process MUST be deterministic, at least for some specific and agreed selection algorithms.

NOTE: Resources used internally by a system for optimization reasons do not have to be exposed through the information system.

NOTE: This document has been agreed from a technical point of view by the developers of the storage services, who have committed to produce information providers that reflect the specifications in this document.

NOTE: This document has also been agreed by client developers who have committed to query the information system accordantly to these specifications.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#).

Document organization

This document is organized in two main sections: the first one covers the details concerning the CPU installed capacity, while the second one tackles the issues concerning storage resources.

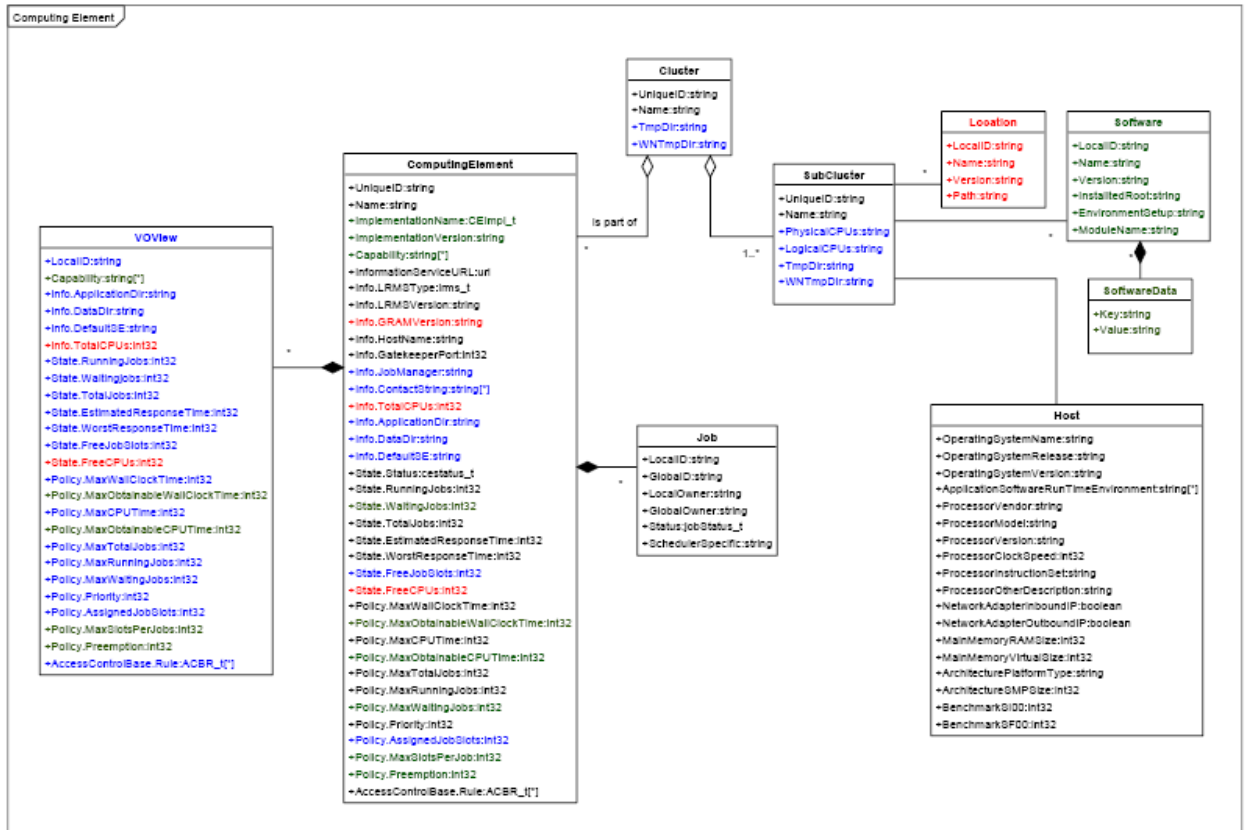


Figure 1: The Glue Schema v1.3 classes for computing resources



1. WLCG Computing Resources

In what follows, we detail the meaning of the several Glue Classes concerning computing resources. We specify how the various attributes must be interpreted and give examples.

We will also cover a description of the current YAIM modules in charge of publishing the information about computing resources.

We conclude with hints to the site administrators and with some notes about validation procedures and with some real examples.

1.1 The Glue Cluster, SubCluster, CE Classes

In Figure 1 we show the Glue Classes Cluster, Subcluster, and CE (and related subclasses) which describe the computing resources at a site.

The attributes in black were present in Glue Schema v1.2; the blue attributes have been introduced with version 1.3; the red attributes have been deprecated in v1.3 in the sense that either their meaning has changed or the attribute is unused. However, all attributes are still defined in the schema used in production, and hence are available to be used if required.

We now describe the individual classes.

Typically, computing resources are contributed to a Grid as set of machines locally managed by batch systems such as the Portable Batch System (PBS), Load Sharing Facility (LSF), Sun Grid Engine (SGE) or Condor. All of them use queues to stage the requests and can have sets of policies associated with the queues and/or groups of users. Moreover, they offer scheduling functionalities to impose the fair sharing of the resources among the set of requests.

In the class diagram above the class *Cluster* represents an aggregation of heterogeneous computing resources managed by a batch system. The *Cluster* class has a set of attributes described in section 3.1 of [1]. A *Cluster* has references to its *SubClusters*, to the Computing Elements or queues that it manages, and to the site it is part of.

A *SubCluster* is a set of (homogeneous) computing resources forming part of a Cluster. The attributes of a subcluster are described in section 3.3 of [1].

Note: Please note that although a *SubCluster* is defined as a homogeneous set of WNs, in practice the way the WMS works limits us to publishing one *SubCluster* per *Cluster* (basically the current WMS can't tell the LRMS which *SubCluster* it wants to use). The result is that in general ***SubClusters are heterogeneous***. Sites have to publish some kind of average WN specification as *SubCluster* attributes. The Worker Nodes Working Group proposes to split the system and have separate queues (CEs), e.g. for large memory nodes. Splitting the system to have separate homogeneous queues is the solution that is proposed to be adopted in WLCG.

Note: It is mandatory for WLCG sites to publish Glue *SubCluster* objects and in particular the attributes outlined in what follows.

The *SubCluster* attributes relevant for making public available information about the installed capacity at sites are:



PhysicalCPUs - defined as the “Total number of real CPUs/physical chips in the SubCluster”.

Please note that by the number of real CPUs we **do not** intend the number of cores [3] in a SubCluster, but indeed the number of processors chips [2] installed in the Worker Nodes comprising a SubCluster.

Please note that the PhysicalCPUs **must be a static number**, i.e. it is configured by the admin and does not change if a few nodes are down temporarily.

The PhysicalCPUs numbers should be summed over all WNs in a SubCluster to obtain the total number of CPUs in that SubCluster.

Another WLCG mandatory attribute is:

LogicalCPUs - defined as the “Total number of cores/hyperthreaded CPUs in the SubCluster”

In other words, LogicalCPUs counts the number of computing units seen by the OS on the WNs. Sites typically configure one job slot per logical CPU, but some sites allow more than this (e.g. five jobs per four cores) to allow for the fact that jobs spend some time waiting for i/o. The 1.3 GLUE schema does not allow such an over allocation to be published explicitly.

Please note that again LogicalCPUs is a static number manually configured by the system administrator at a site and does not reflect the dynamic state of the WNs.

A **Host** entity is attached to the SubCluster. It is meant to describe each node in the (homogeneous) SubCluster. The Host attributes are described in section 3.3 of [1]. The Host attributes relevant for this work are:

BenchmarkSI00- defined as the “SpecInt2000 provided by a typical SubCluster Logical CPU”

For heterogeneous SubCluster we propose the following definition for BenchmarkSI00:

BenchmarkSI00 defined as the “Average SpecInt2000 rating per LogicalCPU”

In the case of a homogenous SubCluster, the only change is that the SpecInt rating is provided per core, instead of per chip. In the case of a heterogeneous SubCluster, the associated BenchmarkSI00 MUST be calculated as follows. If A, B and C are sets of homogeneous machines in a SubCluster SC, if we indicate with:

p_A = the power per core for set A

n_A == number of cores in for set A

and we use a similar symbology for sets B and C, the BenchmarkSI00 for the heterogeneous SubCluster SC is:

$$(1) \quad BenchmarkSI00_{SC} = (p_A * n_A + p_B * n_B + p_C * n_C) / (n_A + n_B + n_C)$$



February 2, 2009

Note: This attribute is normally a static value filled in by hand by the system administrator of the site. This is one source of possible errors, since it depends on a manual procedure to ensure the correctness of this information.

Note: The rule established by WLCG to publish the correct value for BenchmarkSI00 is described in the following GOC wiki page (please check in particular the example given): http://goc.grid.sinica.edu.tw/gocwiki/CPU_Benchmarking_HowTo. *Please, note that in this wiki page one is instructed to find the power per job slot (dividing the total score from the HEPIX procedure by the number of job slots). The proposal we make is to find the power per core (dividing the total score by the number of cores in the box).* Another source of information is the following WLCG document: http://lcg.web.cern.ch/LCG/documents/Processor%20performance%20metrics_26may07.pdf.

Note: The HEPIX benchmark working group has recently proposed a new benchmark unit for CPU power. Details can be found in [11]. Sites that decide to publish CPU power using the new benchmark MUST use the *GlueHostBenchmarkSI00* attribute to publish CPU power. In this case, the power MUST be expressed in SpecInt2000 using the scaling factor that can be found in [11]. Furthermore, such sites MUST also publish the attribute *GlueHostProcessorOtherDescription: Benchmark=<value>-HEP-SPEC*, where in <value> they report the CPU power expressed in the new unit.

We also strongly encourage sites to publish (***WLCG sites must publish***) the following Glue attributes, described in section 3.3 of [1]:

- *ProcessorModel*, note that no schema currently exists for these attributes. Please check section A.2 and A.3 to see what values these attributes can assume.
- *ProcessorSpeed*. It is specified in MHz.
- *SMPSize* is defined as the number of Logical CPUs (cores) per WN.
- *ProcessorOtherDescription* (2)
 - This attribute is used to publish the average number of cores per processor.
 - The recommended way to publish such information is as follows: *Cores=<typical number of cores per CPU>*
 - This attribute is also used to publish the CPU power expressed in HEP-SPEC units as follows: *Benchmark=<value>-HEP-SPEC*, where in <value> they report the CPU power expressed in the new unit. In case both number of cores per processor and CPU power expressed in HEP-SPEC units are published, the syntax is as follows: *Cores=<typical number of cores per CPU> [, Benchmark=<value>-HEP-SPEC]*

Other current issues (not directly related to this discussion) concern the values and relationship of the following attributes:

RAMSize described as the “Total physical memory of a WN in the SubCluster”



February 2, 2009

VirtualSize described as the “*Total virtual memory of a WN in the SubCluster*”

There are a lot of discussions about how these attributes should be interpreted and set. The most common position is to agree that the RAMSize refers to the physical memory for the whole WN. In order to estimate how much memory a job will get once it is running on a specific SubCluster WN it is recommended to divide the total physical memory of a WN (RAMSize) by the SMPSize of a WN. For more details, please check GGUS ticket:

https://gus.fzk.de/ws/ticket_info.php?ticket=26298

The *CE* Glue Class is used to describe a batch queue. There are three important attributes in this class that must be published:

AssignedJobSlots - defined as the “*Total number of assigned Job Slots in the queue available at a given moment*”

Note that this attribute replaces the now-deprecated TotalCPUs attribute. The meaning is unchanged but the former name was misleading; firstly because it counts job slots and not physical or logical CPUs, and these are not necessarily equal, and secondly because the number of slots assigned to a queue is not necessarily the total for the entire system.

Also note that the number of job slots is typically overcounted if summed over all CEs, in the sense that many queues will “see” the same set of job slots.

Furthermore, the AssignedJobSlots in the VOView is a MAXIMUM number, i.e. there is no guarantee that at any given moment, VOs will be able to get this number of job slots.

This is a *dynamic number*, i.e. it reflects the current state of the batch system (depending on the info provider).

The following attributes SHOULD be published. Some are mandatory:

- *CECapability: CPUScalingReferenceSI00=<referenceCPU SI00>*. If batch system time limits are internally scaled to a reference CPU, the rating of that reference MUST be published through this attribute using the same units as published in the BenchmarkSI00 attribute. *<referenceCPU SI00>* is the CPU power expressed in SpecInt2000 for which the published GlueCEMaxCPUTimes are valid. Site administrators should guarantee at least this amount of CPU time on a node of the same type. In the case where internal scaling of CPU times for different logical CPU across the Cluster is used then this is the reference CPU. ***If using no internal batch scaling then the reference CPU is the least powerful core in the Cluster.*** This value MUST always be set. To configure internal scaling for TORQUE a site admin should use cputmult and walltmult. The attribute is used by some WLCG VOs (LHCb) to select queues with an appropriate GlueCEMaxCPUTime. Therefore the publication of this attribute is mandatory.
- *CECapability: Share=<VO>:<share>* ; This value is used to express a specific VO share if VO shares are in operation. *<VO>* is a string set to the name of the VO to which the share refers. *<share>* is an integer with values ranging between 1 and 100. In fact, *<share>* expresses a percentage where 100% is your ***total cluster's capacity***. This means that the sum of the share over all WLCG VO supported by the queues in the cluster MUST be less than or equal to 100. It should only be used where true CPU or wall time shares are in operation. The share is not related to the



case where hard limits are enforced on the number of jobs that can be executed by any particular VO. ***If the cluster has no share configuration then this value SHOULD NOT be published.*** For a real example, please refer to section A.3. As an example of share configuration, you can check the configuration of the batch system at RAL: http://www.gridpp.ac.uk/wiki/RAL_Tier1_PBS_Scheduling. **Please note that OSG sites publish the share in the *GlueSite Sponsor* attribute. When such attribute is used, the value of this attribute is a comma or space-separated list of <VO>:<share> entries, where the value of <share> is the same as described above.**

The *VOView* Glue Class is a subclass of the CE, and is used to report the state of that part of a CE allocated to a specific VO (or to a subgroup or role within a VO). This class includes the *AssignedJobSlots* attribute that reports the number of job slots in a Glue Cluster that a given VO can use. Of course, VOs can share the usage of the same CPU resources. However, this attribute can be used to report on the fraction of the available CPU resources which can be used by a VO.

AssignedJobSlots - defined as the “Total number of job slots part of a Computing Element that a VO can use”.

1.2 Computing the total installed computing capacity at a WLCG site

In order to calculate the ***total installed capacity at a WLCG site*** let us define the function $S(WLCGSubCluster)_{Site}$ to be equal to the sum, normalized by 100, of the shares for WLCG VOs for a queue using a specific WLCG SubCluster at a site, if WLCG VOs shares are published and to 1 if WLCG VOs shares are not published for a queue using a specific WLCG SubCluster:

$$S(WLCGSubCluster)_{Site} = \begin{cases} \sum_{WLCG\ VOs} GlueCECapability(Share) / 100 & \text{if WLCG share are published} \\ & \text{through the GlueCECapability attribute (EGEE sites)} \\ \sum_{WLCG\ VOs} GlueSiteSponsor(Share) / 100 & \text{if WLCG share are published} \\ & \text{through the GlueSiteSponsor attribute (OSG sites)} \\ 1 & \text{if WLCG share are not published} \end{cases}$$

Then the total installed capacity at a site is given by the following formula:

$$Total\ Installed\ Computing\ Capacity_{Site}\ (KSI00) = \sum_{WLCG\ SubClusters} GlueHostBenchMarkSI00 * GlueSubClusterLogicalCPUs * \frac{S(WLCGSubCluster)_{Site}}{10^3}$$



where the sum will be executed over all SubClusters used by queues (CEs) which support WLCG VOs.

In order to find out all WLCG SubClusters to consider when computing the Total Installed Computing Capacity at a site, the algorithm to use is the following:

1. From the WLCG official pledges spread sheet, find the site name given the Federation for which accounting information is required.
2. Given a WLCG Site Name (IFCA-LCG2 in the example) find the GlueSiteUniqueID:

```
ldapsearch -h $LCG_GFAL_INFOSYS -x -b "o=grid"
'(&(ObjectClass=GlueSite)(GlueSiteName=IFCA-LCG2))'
GlueSiteUniqueID

# IFCA-LCG2, IFCA-LCG2, local, grid
dn: GlueSiteUniqueID=IFCA-LCG2,Mds-Vo-name=IFCA-LCG2,Mds- Vo-
name=local,o=grid
GlueSiteUniqueID: IFCA-LCG2
```

3. Find all Clusters for this site and get the GlueClusterUniqueID:

```
ldapsearch -h $LCG_GFAL_INFOSYS -x -b "o=grid"
'(&(ObjectClass=GlueCluster)(GlueForeignKey=GlueSiteUniqueID=IFCA-
-LCG2))' GlueClusterUniqueID

# egeece01.ifca.es, IFCA-LCG2, local, grid
dn: GlueClusterUniqueID=egeece01.ifca.es,Mds-Vo-name=IFCA-
LCG2,Mds-Vo-name=local,o=grid
GlueClusterUniqueID: egeece01.ifca.es
```

4. Find all queues which insist on the set of clusters found which serve a WLCG VO and find the associated GlueClusterUniqueID:

```
ldapsearch -h $LCG_GFAL_INFOSYS -x -b "o=grid"
'(&(ObjectClass=GlueCE)(GlueForeignKey=GlueClusterUniqueID=egeece
01.ifca.es)(|(GlueCEAccessControlBaseRule=VOMS:/alice*)(GlueCEAcc
essControlBaseRule=VO:alice)(GlueCEAccessControlBaseRule=VOMS:/at
las*)(GlueCEAccessControlBaseRule=VO:atlas)(GlueCEAccessControlBa
seRule=VOMS:/cms*)(GlueCEAccessControlBaseRule=VO:cms)(GlueCEAcce
ssControlBaseRule=VOMS:/lhcb*)(GlueCEAccessControlBaseRule=VO:lhcb
b)))' GlueForeignKey

# egeece01.ifca.es:2119/jobmanager-lcgpbs-cms, IFCA-LCG2, local,
grid
dn: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-
cms,Mds-Vo-name=IFCA-LCG2,Mds-Vo-name=local,o=grid
GlueForeignKey: GlueClusterUniqueID=egeece01.ifca.es
```



February 2, 2009

5. For each SubCluster part of the set of Clusters, sum up the product of the published `GlueSubClusterBenchmarkSI00` by `GlueSubClusterLogicalCPUs` and divide the obtained result by 10^3 .

```
ldapsearch -h $LCG_GFAL_INFOSYS -x -b "o=grid"  
'(&(ObjectClass=GlueSubCluster)(GlueChunkKey=GlueClusterUniqueID=  
egeece01.ifca.es))' GlueSubClusterLogicalCPUs  
GlueHostBenchmarkSI00
```

1.2 Validation procedures

It will be necessary to validate the information published by sites against the reality of the resources they are providing to WLCG.

There are numerous possibilities:

1. GStat [7]: Traditionally GStat has been responsible for checking that required attributes are published. New attributes such as *GlueHostProcessorOtherDescription: Cores=<typical number of cores per CPU>* can be checked for by GStat.
2. Site managers can look at the WLCG centrally collected results and decide if they believe them to be correct. The GridMAP [9] interface is currently the easiest visually to view the collected Installed Capacity.
3. More accurate validation can only be achieved by collecting results from executing jobs on the batch worker nodes present on the Grid. It is expected that in the future commands will be available on WNs (Ref <https://savannah.cern.ch/patch/?2114>) to resolve any given WN to the GlueSubCluster to which it belongs. Subsequent comparison of the physical hardware compared to corresponding GlueSubCluster attributes is then possible. A SAM style test performing this comparison is completely feasible.
4. Also in the future we expect the "WNCM" JobWrapper scripts (<http://www.sysadmin.hep.ac.uk/svn/grid-monitoring/trunk/gcm/>) to be deployed and re-enabled at sites. These allow us to collect metrics at start of every job submitted via a GridInterface to a batch farm. Currently these metrics include operating system name, number of cores and other GlueSubCluster attributes. Over time a large amount of data will be collected from every batch worker present on the Grid allowing for comparisons between the reality of the WN sets and what is published into Glue.

Computing Resources Appendix

A.1 Current configuration information and issues for computing resources

The current known problems in reporting correct values for computing resources and their usage are the following:

- a. YAIM: when a new head node is added at a site, YAIM configures automatically a new Cluster and SubCluster associated, causing double counting of resources. In order to avoid double counting of resources when computing the installed capacity at a site, **site administrators MUST**



set the `GlueSubClusterPhysicalCPUs` and `GlueSubClusterLogicalCPUs` attribute to 0 if the resources used by the added SubClusters are already published.

- b. For more information about how current systems like GStat [7] or GridMAP [9] calculate the total number of physical CPUs at a site, please refer to [4] and to GGUS ticket: https://gus.fzk.de/ws/ticket_info.php?ticket=36890

A.2 YAIM CE configuration module

The YAIM module in charge of the configuration of an lcg-CE is `yaim-lcg-ce` and can be found in the gLite CVS:

<http://glite.cvs.cern.ch/cgi-bin/glite.cgi/org.glite.yaim.lcg-ce>

Among other CE services, it configures the generic information provider (GIP) for a Computing Element (CE) node.

Among others, this YAIM module requires the following YAIM variables to be defined:

```
CE_HOST SE_LIST JOB_MANAGER CE_BATCH_SYS VOS QUEUES __GROUP_ENABLE
CE_BATCH_SYS CE_CPU_MODEL CE_CPU_VENDOR CE_CPU_SPEED CE_OS
CE_OS_RELEASE CE_OS_VERSION CE_MINPHYSMEM CE_MINVIRTMEM
CE_SMP_SIZE CE_SIO0 CE_SF00 CE_OUTBOUNDIP CE_INBOUNDIP
CE_RUNTIMEENV INSTALL_ROOT CE_PHYSCPU CE_LOGCPU CE_OS_ARCH
```

Such variables have to be manually set by the system administrator. In what follows we list the correspondence of these variables with those from the Glue Schema v1.3 outlined above.

<code>CE_HOST</code>	This YAIM module configures one <i>GlueCluster</i> and one <i>GlueSubCluster/GlueHost</i> object per Computing Element node. The hostname of the CE node set in the YAIM variable <code>CE_HOST</code> is used as <i>GlueClusterUniqueID</i> , as the host part of the <i>GlueCEUniqueID</i> and as <i>GlueSubClusterUniqueID</i>
<code>CE_PHYSCPU</code>	This YAIM variable is used to set the <i>GlueSubClusterPhysicalCPUs</i> . System administrators MUST set this variable to the value of the “Total number of real CPUs/physical chips in the SubCluster, including the nodes part of the SubCluster that are temporary down or offline”. In order to overcome the current YAIM limitation (see section A.1) when a new CE head node insisting on the same batch resources is added to a site, site admins MUST set the <code>CE_PHYSCPU</code> YAIM variable to 0 if the resources used by the new subclusters are already published via another CE.
<code>CE_LOGCPU</code>	This YAIM variable is used to set the <i>GlueSubClusterLogicalCPUs</i> . System administrators MUST set this variable to the value of the “Total number of cores/hyperthreaded CPUs in the SubCluster, including the nodes part of the SubCluster that are temporary down or offline”. In order to overcome the current YAIM limitation (see section A.1) when a new CE head node insisting on the same batch resources is added to a site, site admins MUST set the <code>CE_LOGCPU</code> YAIM variable to 0 if the resources used by the new subclusters are already published via another CE.



The following variables are used to fill in the *GlueHost* object for a given SubCluster.

CE_OS	This YAIM variable is used to set the <i>GlueHostOperatingSystemName</i> attribute. System administrators MUST set this variable to the name of the operating system used on the Worker Nodes part of the SubCluster.
CE_OS_RELEASE	This YAIM variable is used to set the <i>GlueHostOperatingSystemRelease</i> attribute. System administrators MUST set this variable to the release of the operating system used on the Worker Nodes part of the SubCluster.
CE_OS_VERSION	This YAIM variable is used to set the <i>GlueHostOperatingSystemVersion</i> attribute. System administrators MUST set this variable to the version of the operating system used on the Worker Nodes part of the SubCluster.

For more details and guidance on the values that CE_OS, CE_OS_RELEASE and CE_OS_VERSION can assume, please refer to the following wiki page:

http://goc.grid.sinica.edu.tw/gocwiki/How_to_publish_the_OS_name

CE_CPU_MODEL	This YAIM variable is used to set the <i>GlueHostProcessorModel</i> attribute. System administrators MUST set this variable to the name of the processor model as defined by the vendor for the Worker Nodes in a SubCluster. Given the fact that SubClusters can be heterogeneous, this refers to the <i>typical</i> processor model for the nodes of a SubCluster.
CE_CPU_VENDOR	This YAIM variable is used to set the <i>GlueHostProcessorVendor</i> attribute. System administrators MUST set this variable to the name of the processor vendor for the Worker Nodes in a SubCluster. Given the fact that SubClusters can be heterogeneous, this refers to the <i>typical</i> processor for the nodes of a SubCluster.
CE_CPU_SPEED	This YAIM variable is used to set the <i>GlueHostProcessorClockSpeed</i> attribute. System administrators MUST set this variable to the name of the processor clock speed expressed in MHz for the Worker Nodes in a SubCluster. Given the fact that SubClusters can be heterogeneous, this refers to the <i>typical</i> processor for the nodes of a SubCluster. If you need to publish this value correctly, you are requested to split your CE/Subclusters to be homogenous.
CE_MINPHYSMEM	This YAIM variable is used to set the <i>GlueHostMainMemoryRAMSize</i> attribute. System administrators MUST set this variable to the <i>Total physical memory of a WN in the SubCluster</i> expressed in MegaBytes. Given the fact that SubClusters can be heterogeneous, this refers to the <i>typical</i> worker node in a SubCluster. It is advisable to publish here the minimum total physical memory of the WNs in the SubCluster expressed in MegaBytes. If you need to publish this value correctly, you are requested to split your CE/Subclusters to be homogenous.



February 2, 2009

- CE_MINVIRTMEM This YAIM variable is used to set the *GlueHostMainMemoryVirtualSize* attribute. System administrators MUST set this variable to the *Total virtual memory of a WN in the SubCluster* expressed in MegaBytes. Given the fact that SubClusters can be heterogeneous, this refers to the *typical* worker node in a SubCluster. It is advisable to publish here the minimum total virtual memory of the WNs in the SubCluster expressed in MegaBytes. If you need to publish this value correctly, you are requested to split your CE/Subclusters to be homogenous.
- CE_OUTBOUNDIP This YAIM variable is used to set the *GlueHostNetworkAdapterOutboundIP* attribute. System administrators MUST set this variable to either FALSE or TRUE to express the permission for direct outbound connectivity for the WNs in the SubCluster, even if limited.
- CE_INBOUNDIP This YAIM variable is used to set the *GlueHostNetworkAdapterInboundIP* attribute. System administrators MUST set this variable to either FALSE or TRUE to express the permission for inbound connectivity for the WNs in the SubCluster, even if limited.
- CE_OS_ARCH This YAIM variable is used to set the *GlueHostArchitecturePlatformType* attribute. System administrators MUST set this variable to the Platform Type of the WN in the SubCluster. Given the fact that SubClusters can be heterogeneous, this refers to the *typical* worker node in a SubCluster. More information can be found here: http://goc.grid.sinica.edu.tw/gocwiki/How_to_publish_my_machine_architecture
- CE_SMP_SIZE This YAIM variable is used to set the *GlueHostArchitectureSMPSize* attribute. System administrators MUST set this variable to the number of Logical CPUs (cores) of the WN in the SubCluster. Given the fact that SubClusters can be heterogeneous, this refers to the *typical* worker node in a SubCluster. If you need to publish this value correctly, you are requested to split your CE/Subclusters to be homogenous.
- CE_SI00 This YAIM variable is used to set the *GlueHostBenchmarkSI00* attribute. System administrators MUST set this variable as indicated in (1).
- CE_SF00 YAIM variable is used to set the *GlueHostBenchmarkSF00* attribute.

The following YAIM variables MUST be set as well:

- CE_OTHERDESCR This YAIM variable is used to set the *GlueHostProcessorOtherDescription* attribute. The value of this variable MUST be set to: **Cores=<typical number of cores per CPU> [, Benchmark=<value>-HEP-SPEC]** where <typical number of cores per CPU> is equal to the number of cores per CPU of a typical Worker Node in a SubCluster. The second value of this attribute MUST be published only in the case the CPU power of the SubCluster is computed using the new unit described in [11].
- CE_CAPABILITY This YAIM variable is a blank separated list and is used to set the *GlueCECapability* attribute. In particular, site administrators MUST define the following values: **CPUScalingReferenceSI00=<referenceCPU SI00>**; the reference CPU SI00 is the internal batch scaling factor used to normalize the



February 2, 2009

GlueCEMaxCPUTime. The <referenceCPU SI00> is expressed in SI00. If internal scaling is not done this capability MUST be published and its value set to the minimum value of the corresponding SubClusters GlueHostBenchmarkSI00.

Share=<VO>:<share>; this value is used to express specific VO shares if set. If there is no special share, this value MUST NOT be published. <VO> is the VO name and <share> can assume values between 1 and 100 (it represents a percentage). Please note that the sum of the shares over all WLCG VOs MUST BE less than or equal to 100. For further details, please check the definition of the Share attribute in section 1.1. For a practical example, please check section 1.6.

A.3 Some examples

Here we list an example of what the WLCG Tier-2 IFCA-LCG2 should publish, given the computing resources installed at this site.

Site resources

At the moment this site publishes one Cluster and three homogeneous SubClusters. There are 8 queues (or GlueCE objects) insisting on the same Cluster. From the information obtained by the site administrators these queues insist on the entire Cluster.

The site publishes three homogeneous SubClusters: CMS, INGRID and EIFCA. SubClusters differ for the number of cores per worker node, the physical memory installed and the processor clock speed. Please note that because of the limitation outlined in the first note of section 1.1, with the configuration and information published by the IFCA-LCG2 site, the WMS cannot select the correct computing resources to send a job to in case a JDL requirement is set (for instance to select the needed RAM size for a job). In order to allow the WMS to correctly select the needed resources to run a job, the site administrators should create and publish homogeneous queues associated to each of the SubClusters. CMS is a SubCluster with 28 8-cores Worker Nodes (the number of CPUs per WN is at the moment unknown). INGRID is a SubCluster including 17 4-cores WorkerNodes. EIFCA is a SubCluster comprising 64 2-cores Worker Nodes which were all offline for an upgrade at the moment we made this exercise.

In what follows we show what the site publishes at the moment in black, we show in red the corrected values for some attributes published wrongly and in blue the missing attributes. The same comments made for the CMS SubCluster applies for the other 2 SubClusters. For INGRID and EIFCA we just report the result of the query to the BDII without outlining corrections.

Glue Cluster

```
# egeece01.ifca.es, IFCA-LCG2, local, grid
dn: GlueClusterUniqueID=egeece01.ifca.es,Mds-Vo-name=IFCA-LCG2,Mds-Vo-
name=local,ou=grid
objectClass: GlueClusterTop
objectClass: GlueCluster
objectClass: GlueInformationService
objectClass: GlueKey
```



February 2, 2009

```
objectClass: GlueSchemaVersion
GlueClusterName: egeece01.ifca.es
GlueClusterService: egeece01.ifca.es:2119/jobmanager-lcgpbs-dteam
GlueClusterService: egeece01.ifca.es:2119/jobmanager-lcgpbs-ops
GlueClusterService: egeece01.ifca.es:2119/jobmanager-lcgpbs-cms
GlueClusterService: egeece01.ifca.es:2119/jobmanager-lcgpbs-planck
GlueClusterService: egeece01.ifca.es:2119/jobmanager-lcgpbs-biomed
GlueClusterService: egeece01.ifca.es:2119/jobmanager-lcgpbs-swetest
GlueClusterService: egeece01.ifca.es:2119/jobmanager-lcgpbs-fusion
GlueClusterService: egeece01.ifca.es:2119/jobmanager-lcgpbs-dorii
GlueClusterUniqueID: egeece01.ifca.es
GlueForeignKey: GlueSiteUniqueID=IFCA-LCG2
GlueForeignKey: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-
dteam
GlueForeignKey: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-ops
GlueForeignKey: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-cms
GlueForeignKey: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-
planck
GlueForeignKey: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-
biomed
GlueForeignKey: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-
swetest
GlueForeignKey: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-
fusion
GlueForeignKey: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-
dorii
GlueInformationServiceURL: ldap://egeece01.ifca.es:2170/mds-vo-
name=resource,o=grid
GlueSchemaVersionMajor: 1
GlueSchemaVersionMinor: 3
```

Glue SubClusters

```
# CMS, egeece01.ifca.es, IFCA-LCG2, local, grid
dn: GlueSubClusterUniqueID=CMS,GlueClusterUniqueID=egeece01.ifca.es,Mds-
Vo-name=IFCA-LCG2,Mds-Vo-name=local,o=grid
objectClass: GlueClusterTop
objectClass: GlueSubCluster
objectClass: GlueHostApplicationSoftware
objectClass: GlueHostArchitecture
objectClass: GlueHostBenchmark
objectClass: GlueHostMainMemory
objectClass: GlueHostNetworkAdapter
objectClass: GlueHostOperatingSystem
objectClass: GlueHostProcessor
objectClass: GlueInformationService
objectClass: GlueKey
objectClass: GlueSchemaVersion
```



February 2, 2009

```
GlueChunkKey: GlueClusterUniqueID=egeece01.ifca.es
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_1_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_1_1
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_2_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_3_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_3_1
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_4_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_5_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_6_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_7_0
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_0
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_1
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_2
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_1_0
GlueHostApplicationSoftwareRunTimeEnvironment: R-GMA
GlueHostApplicationSoftwareRunTimeEnvironment: MPI-START
GlueHostApplicationSoftwareRunTimeEnvironment: MPICH
GlueHostApplicationSoftwareRunTimeEnvironment: MPICH-1.2.6
GlueHostApplicationSoftwareRunTimeEnvironment: OPENMPI
GlueHostApplicationSoftwareRunTimeEnvironment: OPENMPI-1.2.2
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-Gate3.0.0-JR
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-Gate3.0.0-LCG
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-TestU01-DISTRNG
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_6_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_7_7
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_8_4
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_0_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_10
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_11
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_4
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_7
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_8
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_9
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-slc3_ia32_gcc323
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-slc4_ia32_gcc345
GlueHostApplicationSoftwareRunTimeEnvironment: VO-planck-XMM-SAS7
GlueHostArchitectureSMPSize: 8
```

The number published for SMPSize is correct and it is equal to the number of cores per worker node. In fact, if we get the file /proc/cpuinfo from one of the worker nodes, we see that:

The number of virtual processors for a worker node in this cluster is given by:

```
grep ^processor /proc/cpuinfo | wc -l
```

which in our case returns 8, that is exactly the number published in SMPSize.



GlueHostArchitecturePlatformType: i686
GlueHostBenchmarkSF00: 2006
GlueHostBenchmarkSI00: ~~1840~~ **1641**

The value published by this site for GlueHostBenchmarkSI00 appears to be wrong. In fact, from values published for GlueHostOperatingSystemName= ScientificCERNSLC, GlueHostOperatingSystemRelease=4.7, GlueHostProcessorClockSpeed=2300 (MHz), GlueHostProcessorModel=Xeon, GlueHostProcessorVendor=Intel, and GlueHostArchitecturePlatformTyp= i686 and given the instructions in the following wiki pages:

http://goc.grid.sinica.edu.tw/gocwiki/How_to_publish_my_machine_architecture

it looks like a typical WN in this SubCluster runs Linux 4 32-bit OS and can be described by the following entry in the CASPUR [8] table:

CPU	Clock speed (GHz)	L2 Cache size (KB)	# cores per box	Memory size (GB)	SPECint_base2000 per box H: high, L: low opt.
Intel Xeon E5345 (Clovertown)	2.33	4096 p. 2 cores	8	16 (FB-DDR2-677)	H: 11230 L: 8753
Barebone: Supermicro CSE-812L-520CB, Mainboard: Supermicro X7DBE, 2 IDE disks					

If we follow the algorithm described in the following wiki to calculate the SI00 for your box:
http://goc.grid.sinica.edu.tw/gocwiki/CPU_Benchmarking_HowTo

we get:

$$\text{BenchMarkSI00} = 8753 * 1.5 / 9 (\text{=number of job slots defined per machine obtained running a job on the machine}) = 1459$$

If we follow our new proposal the number MUST be:

$$\text{BenchMarkSI00} = 8753 * 1.5 / 8 (\text{=number of cores per machine}) = 1641$$

GlueHostMainMemoryRAMSize: 16243
GlueHostMainMemoryVirtualSize: ~~16243~~ **18290**

Concerning Physical and Virtual memory, if we run the command "pbsnodes -a" on this SubCluster, we see:
totmem=18729872kb,availmem=17568592kb,physmem=16633400kb



February 2, 2009

Therefore the correct values to publish are:
GlueHostMainMemoryRAMSize: 16243 (=physmem/1024)
GlueHostMainMemoryVirtualSize: 18290 (=totmem/1024)

```
GlueHostNetworkAdapterInboundIP: FALSE
GlueHostNetworkAdapterOutboundIP: TRUE
GlueHostOperatingSystemName: ScientificCERNSLC
GlueHostOperatingSystemRelease: 4.7 4.6
GlueHostOperatingSystemVersion: SL Beryllium
```

The attributes above are published incorrectly. In fact, following the instructions in: http://goc.grid.sinica.edu.tw/gocwiki/How_to_publish_the_OS_name if we executing /usr/bin/lsc_release -a on a typical WN in this SubCluster we obtain:

```
LSB Version: :core-3.0-ia32:core-3.0-noarch:graphics-3.0-ia32:graphics-3.0-noarch
Distributor ID: ScientificCERNSLC
Description: Scientific Linux CERN SLC release 4.6 (Beryllium)
Release: 4.6
Codename: Beryllium
```

Therefore, the correct values to publish are:
GlueHostOperatingSystemName: ScientificCERNSLC
GlueHostOperatingSystemRelease: 4.6
GlueHostOperatingSystemVersion: Beryllium

```
GlueHostProcessorClockSpeed: 2300
GlueHostProcessorModel: Xeon
GlueHostProcessorVendor: Intel
```

```
GlueSubClusterName: CMS
GlueSubClusterUniqueID: CMS
GlueSubClusterPhysicalCPUs: 224 56
```

The number of physical processors for a worker node is given by:

```
grep 'physical id' /proc/cpuinfo | sort | uniq | wc -l
```

which in our case returns 2.

In this SubCluster there are 28 Worker Nodes. If we multiply 2 by the number of worker nodes in the SubCluster we obtain 56, which is the number that should be published for the attribute PhysicalCPUs.

```
GlueHostProcessorOtherDescription: Cores=4
```



Through the following command we also find the number of cores per physical CPUs:

```
grep 'cpu cores' /proc/cpuinfo
```

```
cpu cores      : 4
cpu cores      : 4
cpu cores      : 4
cpu cores      : 4
cpu cores      : 4
cpu cores      : 4
cpu cores      : 4
cpu cores      : 4
```

GlueSubClusterLogicalCPUs: ~~252~~ **224**

In this SubCluster there are 28 Worker Nodes, each with 8 cores. Therefore, the value of the GlueSubClusterLogicalCPUs MUST be equal to 224.

```
GlueSubClusterTmpDir: /tmp
GlueSubClusterWNTmpDir: /tmp
GlueInformationServiceURL: ldap://egeece01.ifca.es:2170/mds-vo-
name=resource,o=grid
GlueSchemaVersionMajor: 1
GlueSchemaVersionMinor: 3

# INGRID, egeece01.ifca.es, IFCA-LCG2, local, grid
dn:
GlueSubClusterUniqueID=INGRID,GlueClusterUniqueID=egeece01.ifca.es,Mds-Vo-
name=IFCA-LCG2,Mds-Vo-name=local,o=grid
objectClass: GlueClusterTop
objectClass: GlueSubCluster
objectClass: GlueHostApplicationSoftware
objectClass: GlueHostArchitecture
objectClass: GlueHostBenchmark
objectClass: GlueHostMainMemory
objectClass: GlueHostNetworkAdapter
objectClass: GlueHostOperatingSystem
objectClass: GlueHostProcessor
objectClass: GlueInformationService
objectClass: GlueKey
objectClass: GlueSchemaVersion
GlueChunkKey: GlueClusterUniqueID=egeece01.ifca.es
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_1_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_1_1
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_2_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_3_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_3_1
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_4_0
```



February 2, 2009

```
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_5_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_6_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_7_0
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_0
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_1
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_2
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_1_0
GlueHostApplicationSoftwareRunTimeEnvironment: R-GMA
GlueHostApplicationSoftwareRunTimeEnvironment: MPI-START
GlueHostApplicationSoftwareRunTimeEnvironment: MPICH
GlueHostApplicationSoftwareRunTimeEnvironment: MPICH-1.2.6
GlueHostApplicationSoftwareRunTimeEnvironment: OPENMPI
GlueHostApplicationSoftwareRunTimeEnvironment: OPENMPI-1.2.2
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-Gate3.0.0-JR
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-Gate3.0.0-LCG
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-TestU01-DistrNG
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_6_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_7_7
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_8_4
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_0_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_10
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_11
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_4
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_7
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_8
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_9
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-slc3_ia32_gcc323
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-slc4_ia32_gcc345
GlueHostApplicationSoftwareRunTimeEnvironment: VO-planck-XMM-SAS7
GlueHostArchitectureSMPSize: 4
GlueHostArchitecturePlatformType: i686
GlueHostBenchmarkSF00: 2006
GlueHostBenchmarkSI00: 1840
GlueHostMainMemoryRAMSize: 4048
GlueHostMainMemoryVirtualSize: 4048
GlueHostNetworkAdapterInboundIP: FALSE
GlueHostNetworkAdapterOutboundIP: TRUE
GlueHostOperatingSystemName: ScientificCERNSLC
GlueHostOperatingSystemRelease: 4.7
GlueHostOperatingSystemVersion: SL
GlueHostProcessorClockSpeed: 3200
GlueHostProcessorModel: Xeon
GlueHostProcessorVendor: Intel
GlueSubClusterName: INGRID
GlueSubClusterUniqueID: INGRID
GlueSubClusterPhysicalCPUs: 96
GlueSubClusterLogicalCPUs: 116
GlueSubClusterTmpDir: /tmp
```



February 2, 2009

```
GlueSubClusterWNTmpDir: /tmp
GlueInformationServiceURL: ldap://egeece01.ifca.es:2170/mds-vo-
name=resource,o
    =grid
GlueSchemaVersionMajor: 1
GlueSchemaVersionMinor: 3

# EIFCA, egeece01.ifca.es, IFCA-LCG2, local, grid
dn: GlueSubClusterUniqueID=EIFCA,GlueClusterUniqueID=egeece01.ifca.es,Mds-
Vo-n
    ame=IFCA-LCG2,Mds-Vo-name=local,o=grid
objectClass: GlueClusterTop
objectClass: GlueSubCluster
objectClass: GlueHostApplicationSoftware
objectClass: GlueHostArchitecture
objectClass: GlueHostBenchmark
objectClass: GlueHostMainMemory
objectClass: GlueHostNetworkAdapter
objectClass: GlueHostOperatingSystem
objectClass: GlueHostProcessor
objectClass: GlueInformationService
objectClass: GlueKey
objectClass: GlueSchemaVersion
GlueChunkKey: GlueClusterUniqueID=egeece01.ifca.es
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_1_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_1_1
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_2_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_3_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_3_1
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_4_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_5_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_6_0
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2_7_0
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_0
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_1
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_0_2
GlueHostApplicationSoftwareRunTimeEnvironment: GLITE-3_1_0
GlueHostApplicationSoftwareRunTimeEnvironment: R-GMA
GlueHostApplicationSoftwareRunTimeEnvironment: MPI-START
GlueHostApplicationSoftwareRunTimeEnvironment: MPICH
GlueHostApplicationSoftwareRunTimeEnvironment: MPICH-1.2.6
GlueHostApplicationSoftwareRunTimeEnvironment: OPENMPI
GlueHostApplicationSoftwareRunTimeEnvironment: OPENMPI-1.2.2
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-Gate3.0.0-JR
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-Gate3.0.0-LCG
GlueHostApplicationSoftwareRunTimeEnvironment: VO-biomed-TestU01-DistrNG
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_6_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_7_7
```



February 2, 2009

```
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_1_8_4
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_0_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_10
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_11
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_12
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_4
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_7
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_8
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-CMSSW_2_1_9
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-slc3_ia32_gcc323
GlueHostApplicationSoftwareRunTimeEnvironment: VO-cms-slc4_ia32_gcc345
GlueHostApplicationSoftwareRunTimeEnvironment: VO-planck-XMM-SAS7
GlueHostArchitectureSMPSize: 2
GlueHostArchitecturePlatformType: i686
GlueHostBenchmarkSF00: 2006
GlueHostBenchmarkSI00: 1840
GlueHostMainMemoryRAMSize: 2048
GlueHostMainMemoryVirtualSize: 2048
GlueHostNetworkAdapterInboundIP: FALSE
GlueHostNetworkAdapterOutboundIP: TRUE
GlueHostOperatingSystemName: ScientificCERNSLC
GlueHostOperatingSystemRelease: 4.7
GlueHostOperatingSystemVersion: SL
GlueHostProcessorClockSpeed: 3200
GlueHostProcessorModel: Xeon
GlueHostProcessorVendor: Intel
GlueSubClusterName: EIFCA
GlueSubClusterUniqueID: EIFCA
GlueSubClusterPhysicalCPUs: 160
GlueSubClusterLogicalCPUs: 250
GlueSubClusterTmpDir: /tmp
GlueSubClusterWNTmpDir: /tmp
GlueInformationServiceURL: ldap://egeece01.ifca.es:2170/mds-vo-
name=resource,o
=grid
GlueSchemaVersionMajor: 1
GlueSchemaVersionMinor: 3
```

Glue CEs

In what follows we show the configuration of one batch queue only, the one with GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-cms.

```
# egeece01.ifca.es:2119/jobmanager-lcgpbs-cms, IFCA-LCG2, local, grid
dn: GlueCEUniqueID=egeece01.ifca.es:2119/jobmanager-lcgpbs-cms,Mds-Vo-
name=IFCA-LCG2,Mds-Vo-name=local,o=grid
objectClass: GlueCETop
objectClass: GlueCE
objectClass: GlueCEAccessControlBase
```



February 2, 2009

```
objectClass: GlueCEInfo
objectClass: GlueCEPolicy
objectClass: GlueCEState
objectClass: GlueInformationService
objectClass: GlueKey
objectClass: GlueSchemaVersion
GlueCEHostingCluster: egeece01.ifca.es
GlueCEName: cms
GlueCEUniqueID: egeece01.ifca.es:2119/jobmanager-lcgpbs-cms
GlueCEImplementationName: LCG-CE
GlueCEImplementationVersion: 31
GlueCECapability: unset
```

The CPUScalingFactorSI00 attribute MUST always be published. If the site uses a batch system CPU scaling factor to set the GlueCEPolicyMaxCPUTime the reference CPU must be published as follows. In the example we assume that the reference CPU is a P4. Please check the following twiki page for further instructions:

http://goc.grid.sinica.edu.tw/gocwiki/Is_there_a_set_of_reference_CPUs

GlueCECapability: CPUScalingFactorSI00=852

Also the Share attribute MUST be published if a share is defined for a specific VO. If a specific share is applied for a VO, the share percentage for the VO MUST BE manually set as follows. In this example we assume that CMS has a share of 60% over all Cluster resources, biomed has a 20%, plunk has 10%, fusion has 2%, dorii has 2%, swetest has 2%, dteam 2% and ops 2%. We express that using the GlueCECapability attribute for each defined queue. Please note that **the sum of the shares over all WLCG VOs MUST BE less than or equal to 100**. Furthermore, the same shares are published multiple times, one for each CE queue, even if the queue cannot be accessed by the other VOs (like for this CE in the example, only accessible by CMS).

```
GlueCECapability: Share=cms:60
GlueCECapability: Share=biomed:20
GlueCECapability: Share=plunk:10
GlueCECapability: Share=fusion:2
GlueCECapability: Share=dorii:2
GlueCECapability: Share=swetest:2
GlueCECapability: Share=dteam:2
GlueCECapability: Share=ops:2
```

If no share is defined, than these values MUST NOT be published.

```
GlueCEInfoGatekeeperPort: 2119
GlueCEInfoHostName: egeece01.ifca.es
GlueCEInfoLRMSType: torque
GlueCEInfoLRMSVersion: 2.3.0-snap.200801151629
GlueCEInfoJobManager: lcgpbs
GlueCEInfoContactString: gram://egeece01.ifca.es:2119/jobmanager-lcgpbs
GlueCEInfoApplicationDir: /opt/exp_soft
```



February 2, 2009

```
GlueCEInfoDataDir: unset
GlueCEInfoDefaultSE: egeese01.ifca.es
GlueCEInfoTotalCPUs: 337 292
```

We asked the site administrator at IFCA-LCG2 and they told us that the cms queue can use all available CPUs in the cluster. On each WN the number of slots is equal to the number of cores + 1. This extra slot is only used for monitoring purposes and cannot be used by real jobs. Therefore, the number of TotalCPUs or AssignedJobSlots MUST be the same as the total number of active cores in the cluster.

Given that in the CMS SubCluster there are 224 active cores, in the INGRID SubCluster there are 68 active cores and in the EIFCA SubClusters there are 128 cores which are offline, the number that MUST be published as GlueCEInfoTotalCPUs is 420-128 which is equal to 292.

Please note that this attribute is deprecated. Therefore, the GlueCEPolicyAssignedJobSlots attribute should be set instead.

```
GlueCEStateEstimatedResponseTime: 5295
GlueCEStateRunningJobs: 223
GlueCEStateStatus: Production
GlueCEStateTotalJobs: 263
GlueCEStateWaitingJobs: 40
GlueCEStateWorstResponseTime: 10368000
GlueCEStateFreeJobSlots: 0
```

The value of this attribute is correctly set to 0 since there are waiting jobs and the EstimatedResponseTime (ERT) is non-zero.

```
GlueCEStateFreeCPUs: 110
```

The value of this attribute MUST be the same as the value of GlueCEStateFreeJobSlots. This is due to a bug in the information providers, as outlined in the following savannah bug:

<https://savannah.cern.ch/bugs/index.php?15973>

Please, check also the following two related bugs:

<https://savannah.cern.ch/bugs/index.php?15974>

<https://savannah.cern.ch/bugs/index.php?22963>

```
GlueCEPolicyMaxCPUtime: 3600
```

```
GlueCEPolicyMaxObtainableCPUtime: 0
GlueCEPolicyMaxRunningJobs: 0
GlueCEPolicyMaxWaitingJobs: 0
GlueCEPolicyMaxTotalJobs: 0
```




February 2, 2009

The attributes above MaxXXX are unset and the value of 0 is used to show that they are unset. However, this is not a good practice since this value might imply errors if used in simple arithmetical expressions. They should be set to a very big number (such as 999999999). These attributes are set automatically by YAIM. Please check the following savannah bugs:

<https://savannah.cern.ch/bugs/?44533>

<https://savannah.cern.ch/bugs/?44534>

```
GlueCEPolicyMaxWallClockTime: 4320
GlueCEPolicyMaxObtainableWallClockTime: 0
GlueCEPolicyPriority: 1
GlueCEPolicyAssignedJobSlots: + 292
```

As already pointed out in the comment on the attribute GlueCEInfoTotalCPUs, the value of this attribute should not be 0 but 292.

```
GlueCEPolicyMaxSlotsPerJob: 0
GlueCEPolicyPreemption: 0
GlueCEAccessControlBaseRule: VO:cms
GlueCEAccessControlBaseRule: VOMS:/cms
```

This entry is not needed since the support for the CMS VO has been already published in the previous line. The format is correct since valid formats for the AccessControlBaseRule are:

- a. <DN>
- b. <VO NAME> - deprecated
- c. VO:<VO NAME>
- d. VOMS:<FQAN>

And /cms is a FQAN.

```
GlueCEAccessControlBaseRule: VOMS:/cms/Role=lcgadmin
GlueCEAccessControlBaseRule: VO:cms
```

This entry is a duplication.

```
GlueCEAccessControlBaseRule: VOMS:/cms/Role=production
GlueForeignKey: GlueClusterUniqueID=egeece01.ifca.es
GlueInformationServiceURL: ldap://egeece01.ifca.es:2170/mds-vo-
name=resource,o=grid
GlueSchemaVersionMajor: 1
GlueSchemaVersionMinor: 3
```



2. WLCG Storage Resources

In what follows, we detail the meaning of the several Glue Classes concerning storage resources. We specify how the various attributes MUST be interpreted for WLCG and give examples. The goal is to provide detailed information for storage information provider developers.

2.1 Glue Storage Element, Storage Area, VOInfo, Control and Access Protocol

In Figure 2 we show the Glue Classes describing storage resources.

We now describe each class and its attributes, for the purpose of providing installed and used resource capacity.

The Storage Element (SE) Class describes storage resources at a site. There MAY be more than one SE at one given site. A Storage Element represents a convenient partition of the storage resources of one or more storage systems as a single Grid entity.

The attributes of the Storage Element Class are described in section 4.1 of [1]. Here are some more comments:

- *UniqueID* MUST be set and the value SHOULD be the FQDN of the SRM host, if the storage service has an SRM interface. Clients SHOULD NOT rely on semantics of this attribute. In fact, current clients assume that the Storage Element Unique ID is equal to host part of the correspondent SRM service FQDN (please check section 2.4). Therefore, at the moment we cannot drop this requirement.
- The *Port* attribute of the Storage Element class is deprecated and SHOULD NOT be published.
- The *Name* attribute MAY be published and MAY take any value. Clients MUST NOT rely on semantics of this attribute.
- The *SizeTotal*, *SizeFree*, *TotalOnlineSize* and *UsedOnlineSize* (in GB) SHOULD be published. They SHOULD be aggregated from what's in the SAs and they SHOULD summarize the space in the entire SE. For more details, please check the discussion in GGUS ticket: https://gus.fzk.de/pages/ticket_details.php?ticket=36400. For SEs not supporting tapes, *TotalNearlineSize* and *UsedNearlineSize* MUST be published with a value of 0. It is not mandatory to set the value for the following attributes:
 - *UsedOnlineSize* This attribute SHOULD be published. If published, it MUST provide information in Gigabytes and SHOULD follow the following semantics: an empty StorageElement (one with no files) reports 0 GB; when a file of size x is made available using online resources and under normal operating conditions *UsedOnlineSize* will increase by x in a timely fashion; when the StorageElement no longer guarantees that a file of size x is available from online resources and under normal operating conditions *UsedOnlineSize* will decrease by x in a timely fashion.
 - *TotalOnlineSize* This attribute SHOULD be published. If published, it MUST provide information in Gigabytes and SHOULD provide the aggregated capacity of all available online resources independent of usage. If an online resource known to a Storage Element is temporarily unavailable, it MAY be included in *TotalOnlineSize*.

- The *Status* attribute MUST be published and MUST assume the following values: *Production*, *Closed*, *Draining* (existing transactions can complete but new ones don't start) and *Queueing* (users can start transactions but they won't actually run). The main motivation for introducing this was so a site administrator could set the status to *Draining* for an SE which he/she wants to shut down, rather than just switching off abruptly. However, it may be that it would have to be set explicitly by the administrator. To be useful we would also need support in the clients, but that's waiting on getting the publication right: see <https://savannah.cern.ch/bugs/index.php?31706>

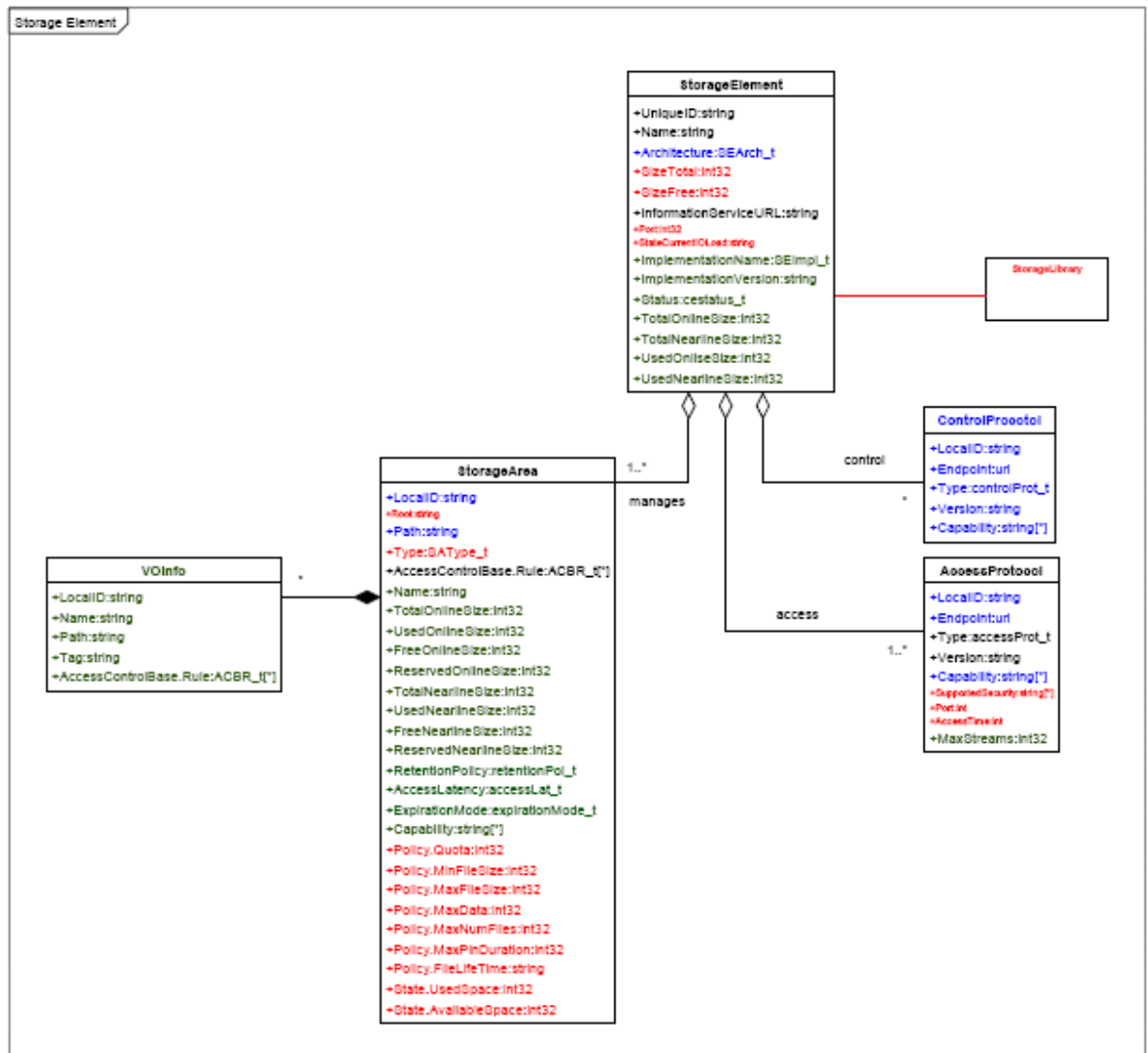


Figure 2: The Glue Schema v1.3 classes for storage resources



February 2, 2009

- The *Architecture* SHOULD be published. If published, its value MUST be one of *disk*, *multidisk*, *tape*, *other*. If publishing *Architecture* an SE providing any tape resources MUST publish *tape*. Otherwise, any kind of RAID array or similar technology MUST be published as *multidisk*; *disk* MUST only be used for systems using individual disks with no redundancy. Other technology e.g. solid-state or optical disks MUST be published as *other*.
- Publishing *InformationServiceURL* is OPTIONAL; we don't require resource BDIIs to be contactable.
- The *CurrentIOLoad* attribute SHOULD NOT be published.
- *ImplementationName* MUST be published. The format is defined by the implementations, but the names MUST NOT clash. In particular, for WLCG storage services the names to be used are: *CASTOR*, *DPM*, *StoRM*, *dCache*, *BestMan*, *xrootd*.
- *ImplementationVersion* MUST be published. The format is defined by the implementations.

The ***Access Protocol*** class describes the file access protocols supported by the Storage Element. Also the attributes of this class are described in section 4.3 of [1]. Here some more refined comments:

- *LocalID* MUST be published. Clients MUST NOT interpret any semantic significance to the *LocalID*.
- *Type* MUST be published. It is defined in: <http://glueschema.forge.cnaf.infn.it/V12/SEAccessProtocolType> . In case the Storage Services supports SRM as control protocol, the *Type* SHOULD correspond to the protocol name used by the SRM in a PrepareToGet/Put operation, but there does not appear to be a definitive list of these names. An implementation providing the secure rfio protocol MUST publish an *AccessProtocol* with *Type* *rfio* and MUST publish *GSI* in *SupportedSecurity* (see below). It is RECOMMENDED to publish an additional ***AccessProtocol*** object where the *Type* is set to *gsirfio* and *SupportedSecurity* is 'GSI' as before. Clients MUST NOT depend on the availability of an *Access Protocol* with type *gsirfio*. For the insecure version of the protocol the *Type* MUST be *rfio* and *SupportedSecurity* MUST be unset.
- The *Port* is made obsolete by the *Endpoint* and MAY be empty.
- The *Endpoint* MAY be empty. If a Storage Element has an SRM interface, the TURL for accessing a file via a given supported protocol is returned by the SRM server.
- The *Version* MUST be published. It MUST be the version of the access protocol, not the implementation.
- The *SupportedSecurity* if set, SHOULD be one of the agreed protocols. At present, the agreed protocols are: *GSI* (for any secure protocol based on GSI) and *tkauthz* (for the *tkauthz* security mechanism used by ALICE). Please note that at the moment nothing uses this attribute.
- *Capability* is undefined at the moment.



- The *MaxStreams* attribute MAY be published. If published, the *MaxStreams* value SHOULD indicate the maximal number of parallel streams the implementation can support.

The *Control Protocol* class describes the control protocols supported by the Storage Element. The attributes of this class are described in section 4.4 of [1]. Here some more refined comments:

An SE MUST publish one or more Control Protocol objects (advertising for instance the “pure” xroot, SRM 2.2 or 1.1 protocols). As things stand, there is no reliable way to connect the GlueSE/SA etc to GlueService: formally the *only* way to know the SE endpoint is from the Control Protocol object. ***Please note that storage services offering only an “xroot door” and not a pure xroot implementation, MUST NOT publish a control protocol object for xroot. Such services MUST only publish an Access Control object for xroot.***

- The *Endpoint* MUST be the full control endpoint for the protocol advertised.
 - For the SRM protocol, the following rules apply:
 - The *Endpoint* URL scheme MUST be the SOAP transfer protocol.
 - A URL port number MUST be published in the *Endpoint* URL.
 - An SE MUST publish a SOAP endpoint supporting GSI over HTTP. For this protocol, the URL scheme MUST be ‘httpg’.

Examples:

the URL: `httpg://srm.example.com:8443/srm/managerv2` is a good Control Protocol Endpoint while the following are wrong examples:

`srm://srm.example.com:` the port is missing port; it is not a full endpoint, the protocol srm is specified instead of httpg

`srm://srm.example.com/srm/managerv2:` the port is missing, the protocol srm is specified instead of httpg

`httpg://srm.example.com:8443/srm/managerv2?SFN=...`: It is a full SOAP endpoint however the additional `?SFN=(...)` string was appended

- For the xroot protocol, the following rules apply:
 - The full control endpoint is of the following form:
`xroot://[user@]host[:port][,host[:port]...]/`
Please, note the ending slash that must be present for the endpoint to be valid.
- *Type* MUST be *SRM* (upper case) for the SRM protocol and *xroot* (lower case) for the xroot protocol.



February 2, 2009

- *Version* MUST be '1.1.0' or '2.2.0' for SRM v1 and v2 respectively. The most recent version of the xroot protocol at the moment of writing is 2.9.4.
- *Capabilities* are undefined at the moment. However, this is would be a good place to advertise specific features which can be optional.

We now describe the GlueSA and GlueVOInfo classes that are the most relevant for the publication of storage resource installation and usage. Here we describe only the attributes discussed by the working group.

The Glue *Storage Area* (SA) class describes a logical view of a portion of physical space that can include disks and tape resources. SAs MAY overlap. Shared portions of storage MUST be represented with a single GlueSA object, with multiple GlueSAAccessControlBaseRule attributes and optionally with multiple VOInfo objects pointing to it.

Normally a Storage Area is used to represent SRM-reserved space. It is RECOMMENDED that a Storage Area object be published for portions of storage configured but yet unreserved. In this case the SA MUST publish ReservedOnlineSize=0 and ReservedNearlineSize=0. A special Capability attribute must be used to describe the total installed capacity (the InstalledCapacity attribute described later). Please note that for overlapping published Storage Areas, in order to avoid double counting of resources, only some of them MUST publish a value for the InstalledCapacity attribute greater than 0 (please, check the details later in this section). The sum of all InstalledCapacity attributes over all Storage Areas of an SE MUST BE equal to the physical storage capacity for that SE. When undertaking storage accounting, clients SHOULD identify end-users of physical hardware from the published SA.ACBR values. Here follows a detailed list of the relevant attributes:

- *LocalID* MUST be published. It is a string that identifies this specific Storage Area within the SE. GLUE does not impose any format. The LocalID MAY encode internal information about the space but clients MUST NOT rely on this. The Local ID SHOULD NOT match the name of the VO using this SA.
- The *SAPath* MUST NOT be published unless there is noone VOInfo publishing VOInfoPath, or unless all VOInfos connected to this SA publish the same path. Please note that the SAPath SHOULD NOT be published if the VOInfoPath is published. The old SARoot attribute is obsolete and MUST NOT be published.
- *AccessControlBaseRule* This attribute SHOULD be set. Formally it is allowed to publish an SA with no ACBRs, although it would be inaccessible - e.g. to publish unallocated space. If this attribute is set, its value MUST be one of the following:
 - a. <DN>
 - b. <VO NAME> - deprecated
 - c. VO:<VO NAME>
 - d. VOMS:<FQAN>

Clients MUST be able to accept multi-valued AccessControlBaseRule. Multiple ACBRs are ORed, i.e. access is assumed to be allowed if any of them match. Clients SHOULD ignore ACBR schemes they do not understand to allow for future expansion. There is NO negative ACBR.



Clients SHOULD be prepared to accept incorrectly formatted AccessControlBaseRule attribute values containing the VO name only, with no qualifying scheme identifier. Wildcards in ACBRs are currently not allowed, but there is an EGEE proposal [6] to support limited wildcards.

- *Reserved[Online/Nearline]Size* (in GB=10⁹ bytes) MUST be published. It is a portion of available storage physically allocated to a VO or to a set of VOs. The value of this attribute MUST be 0 for a Storage Area representing an unreserved space. The Reserved Online [Nearline] Size MUST NOT be negative. For WLCG usage *Online* refers to space on disk while *Nearline* refers to space on tape. For tapes, sizes MUST be reported publishing the actual size on tape after compression.
- *Total[Online/Nearline]Size* (in GB=10⁹ bytes) is the total online or nearline space available at a given moment (it does not include broken disk servers, draining pools, etc.). This attribute MUST be published. In the absence of unavailable pools the Total Size is equal to the Reserved Size if the Reserved Size is greater than 0. The Total Online [Nearline] Size MUST NOT be a negative number and MUST NOT exceed the Reserved Online [Nearline] Size if the Reserved Online [Nearline] Size is greater than 0.
- *Used[Online/Nearline]Size* (in GB=10⁹ bytes) is the space occupied by available and accessible files that are not candidates for garbage collection. This attribute MUST be published. For CASTOR, since all files in T1D0 are candidates for garbage collection, it has been agreed that in this case UsedOnlineSize is equal to GlueSATotalOnlineSize. For T0D1 classes of storage this is the space occupied by valid files. Size MUST NOT be a negative number and MUST NOT exceed the Total Online [Nearline] Size. For a definition of *TnDm*, please refer to [5].
- *Free[Online/Nearline]Size* (in GB=10⁹ bytes) MUST be published and MUST be equal to Total – Used
- *Capability* is a string that publishes various characteristics of a Storage Area. At the moment the agreed capabilities of a Storage Area are:
 - a. *Installed[Online/Nearline]Capacity=<size>* (in GB=10⁹ bytes). This attribute has been added for accounting purposes. This attribute MUST always be published. This expresses the size of the physical space of a Storage Area. Each SA MUST be published with at least two Capability attributes: *InstalledOnlineCapacity=<size>* and *InstalledNearlineCapacity=<size>*; both values for size MUST be non-negative. The sum, over all SAs, of *InstalledOnlineCapacity* values or the sum, over all SAs, of *InstalledNearlineCapacity* values SHOULD add up to the corresponding available physical resources represented by the SE for disk and tape respectively. As a special case, an *InstalledOnlineCapacity* or *InstalledNearlineCapacity* of 0 indicates that the SA MUST be ignored for all disk or tape storage accounting respectively. To give an example of publishing unavailable space in CASTOR, an SA with disk servers online in a service class not associated to a specific space token description for a given VO will be published with the *InstalledOnlineCapacity=<size>* capability.
 - b. *scratch*. This capability indicates that this Storage Area is of the type T0D0, not supported according to the WLCG SRM Usage Agreement. In this space, files MUST be created by users as VOLATILE and MAY be removed by the system as soon as their lifetime expires.



- c. *stage*. This is a Storage Area used for staging operations only. If this attribute is published then at least one member of one of the listed `GlueSAAccessControlBaseRule` end-user groups can use the SA as a staging. This SA has no associated `VOInfo` object. The `SAAccessControlBaseRules` list all FQAN that can use this SA.

The Glue *VOInfo* class describes VO specific attributes of a Storage Area, or better it gives a view of a Storage Area from a VO perspective. One Storage Area can be associated to zero or more `VOInfo` objects. *VOInfo represents the ability of a set of end-users to write into a particular Storage Area with a set of parameters specified by the VOInfo class attributes.*

- *VOInfoLocalID* MUST be published. It is a string that identifies this specific `VOInfo` object within an SA. It does not have any imposed format and clients MUST NOT make any assumptions about it.
- *VOInfoTag*. If an SA contains resources allocated to a VO and the VO can access such an SA via a space token description, then the SA MUST have a `VOInfo` object associated with it which publishes the space token description. The *VOInfoTag* attribute MUST be used for this.
- *VOInfoAccessControlBaseRule* MUST be published. . This MUST correspond to a logical subset of the rules published in the corresponding `SAAccessControlBaseRules`, and the ACBRs for `VOInfo` objects associated with the same SA MUST NOT overlap. The value MUST be one of the following:
 - a. <DN>
 - b. <VO NAME> - deprecated
 - c. VO:<VO NAME>
 - d. VOMS:<FQAN>

Clients MUST be able to accept multi-valued `AccessControlBaseRule`. Multiple ACBRs are ORed, i.e. access is assumed to be allowed if any of them match. Clients SHOULD ignore ACBR schemes they do not understand to allow for future expansion. There is NO negative ACBR. Clients SHOULD be prepared to accept incorrectly formatted `AccessControlBaseRule` attribute values containing the VO name only, with no qualifying scheme identifier. Wildcards in ACBRs are currently not allowed, but there is an EGEE proposal [6] to support limited wildcards.

- *VOInfoPath* MUST be published. It is a string that describes the Path to be used in constructing a SURL when writing to the associated Storage Area. Please note that Storage Areas that can be accessed in read only mode or used for staging only activities by all VOs that can access them do not need to have a `VOInfo` associated object. Please note that the `SAPath` SHOULD NOT be published if there is at least a `VOInfo` object associated to a given Storage Area. If both path attributes are published, the client MUST use the one in the `VOInfo` object.



2.2 Computing the Total Installed Online and Nearline Storage Capacity at a Site

In order to calculate the total storage installed capacity at a site the following formula will be used in WLCG:

$$\begin{aligned} \text{Total Installed Online Storage Capacity}_{\text{Site}} \text{ (GB)} &= \\ (\sum_{\text{WLCG GlueSA}} \text{GlueSACapability(InstalledOnlineCapacity)}) & \\ \text{Total Installed Nearline Storage Capacity}_{\text{Site}} \text{ (GB)} &= \\ (\sum_{\text{WLCG GlueSA}} \text{GlueSACapability(InstalledNearlineCapacity)}) & \end{aligned}$$

where the sum will be executed over all Storage Areas used by WLCG VOs.

To satisfy the use case 2 in the introduction to this document the formula above works as long as the sum is restricted to the GlueSA objects that can be used by a specific VO. In case a GlueSA is shared with other VOs, the total won't be accurate.

To satisfy the use case 3 in the introduction to this document the formula above should not use the *GlueSACapability(Installed[Online/Nearline]Capacity)* but instead the *Used[Online/Nearline]Size* of the GlueSA objects used by a specific VO.

In order to find out all WLCG GlueSA to consider when computing the Total Installed Storage Capacity at a site, the algorithm to use is the following:

1. From the WLCG official pledges spread sheet, find the site name given the Federation for which accounting information is required.
2. Given a WLCG Site Name (GRIF in the example) find the GlueSiteUniqueID:

```
ldapsearch -h $LCG_GFAL_INFOSYS -x -b "o=grid"
'(&(ObjectClass=GlueSite) (GlueSiteName=GRIF))' GlueSiteUniqueID

# GRIF, GRIF-LAL, GRIF, local, grid
dn: GlueSiteUniqueID=GRIF,Mds-Vo-name=GRIF-LAL,Mds-Vo-
name=GRIF,Mds-Vo-name=local,o=grid
GlueSiteUniqueID: GRIF
```

3. Find all Storage Elements for this site and get the GlueSEUniqueID:

```
ldapsearch -h $LCG_GFAL_INFOSYS -x -b "o=grid"
'(&(ObjectClass=GlueSE) (GlueForeignKey=GlueSiteUniqueID=IFCA-
LCG2))' GlueSEUniqueID
```



```
# apcse01.in2p3.fr, GRIF-APC, GRIF, local, grid
dn: GlueSEUniqueID=apcse01.in2p3.fr,Mds-Vo-name=GRIF-APC,Mds-Vo-
name=GRIF,Mds-Vo-name=local,o=grid
GlueSEUniqueID: apcse01.in2p3.fr

# lpnse1.in2p3.fr, GRIF-LPNHE, GRIF, local, grid
dn: GlueSEUniqueID=lpnse1.in2p3.fr,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-
name=GRIF,Mds-Vo-name=local,o=grid
GlueSEUniqueID: lpnse1.in2p3.fr

# ipnsedpm.in2p3.fr, GRIF-IPNO, GRIF, local, grid
dn: GlueSEUniqueID=ipnsedpm.in2p3.fr,Mds-Vo-name=GRIF-IPNO,Mds-Vo-
name=GRIF,Mds-Vo-name=local,o=grid
GlueSEUniqueID: ipnsedpm.in2p3.fr

# grid03.lal.in2p3.fr, GRIF-LAL, GRIF, local, grid
dn: GlueSEUniqueID=grid03.lal.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-
name=GRIF,Mds-Vo-name=local,o=grid
GlueSEUniqueID: grid03.lal.in2p3.fr

# grid05.lal.in2p3.fr, GRIF-LAL, GRIF, local, grid
dn: GlueSEUniqueID=grid05.lal.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-
name=GRIF,Mds-Vo-name=local,o=grid
GlueSEUniqueID: grid05.lal.in2p3.fr

# nodel2.datagrid.cea.fr, GRIF-IRFU, GRIF, local, grid
dn: GlueSEUniqueID=nodel2.datagrid.cea.fr,Mds-Vo-name=GRIF-
IRFU,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSEUniqueID: nodel2.datagrid.cea.fr

# polgrid4.in2p3.fr, GRIF-LLR, GRIF, local, grid
dn: GlueSEUniqueID=polgrid4.in2p3.fr,Mds-Vo-name=GRIF-LLR,Mds-Vo-
name=GRIF,Mds-Vo-name=local,o=grid
GlueSEUniqueID: polgrid4.in2p3.fr
```

4. Find all Storage Areas part of the SEs found that serve WLCG VOs and for those find the GlueSACapability Installed[Online|Nearline]Capacity:

```
ldapsearch -h $LCG_GFAL_INFOSYS -x -b "o=grid"
' (&(ObjectClass=GlueSA) (| (GlueChunkKey=GlueSEUniqueID=apcse01.in2p3
.fr) (GlueChunkKey=GlueSEUniqueID=lpnse1.in2p3.fr) (GlueChunkKey=Glue
SEUniqueID=lpnse1.in2p3.fr) (GlueChunkKey=GlueSEUniqueID=grid03.lal.
in2p3.fr) (GlueChunkKey=GlueSEUniqueID=grid05.lal.in2p3.fr) (GlueChun
kKey=GlueSEUniqueID=nodel2.datagrid.cea.fr) (GlueChunkKey=GlueSEUniq
ueID=polgrid4.in2p3.fr)) (| (GlueSAAccessControlBaseRule=VOMS:/alice*
) (GlueSAAccessControlBaseRule=VOMS:/atlas*) (GlueSAAccessControlBase
Rule=VOMS:/cms*) (GlueSAAccessControlBaseRule=VOMS:/lhcb*) (GlueSAAcc
```



February 2, 2009

```
essControlBaseRule=VO:alice) (GlueSAAccessControlBaseRule=VO:atlas) (
GlueSAAccessControlBaseRule=VO:cms) (GlueSAAccessControlBaseRule=VO:
lhcb)))' GlueSACapability
```

```
# ATLASMCDISK:replica:online, grid05.lal.in2p3.fr, GRIF-LAL, GRIF,
local, grid
dn:GlueSALocalID=ATLASMCDISK:replica:online,GlueSEUniqueID=grid05.l
al.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=16492
```

```
# cms:replica:online, node12.datagrid.cea.fr, GRIF-IRFU, GRIF,
local, grid
dn:GlueSALocalID=cms:replica:online,GlueSEUniqueID=node12.datagrid.
cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=31695
```

```
# ATLASGROUPDISK:replica:online, lpnse1.in2p3.fr, GRIF-LPNHE, GRIF,
local, grid
dn:GlueSALocalID=ATLASGROUPDISK:replica:online,GlueSEUniqueID=lpnse
1.in2p3.fr,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=4398
```

```
# ATLASDATADISK:replica:online, lpnse1.in2p3.fr, GRIF-LPNHE, GRIF,
local, grid
dn:GlueSALocalID=ATLASDATADISK:replica:online,GlueSEUniqueID=lpnse1
.in2p3.fr,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=4398
```

```
# atlas:replica:online, grid05.lal.in2p3.fr, GRIF-LAL, GRIF, local,
grid
dn:GlueSALocalID=atlas:replica:online,GlueSEUniqueID=grid05.lal.in2
p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=12743
```

```
# cms:replica:online, lpnse1.in2p3.fr, GRIF-LPNHE, GRIF, local,
grid
dn:GlueSALocalID=cms:replica:online,GlueSEUniqueID=lpnse1.in2p3.fr,
Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=3783
```

```
# ATLASGROUPDISK:replica:online, grid05.lal.in2p3.fr, GRIF-LAL,
GRIF, local, grid
```



February 2, 2009

```
dn:GlueSALocalID=ATLASGROUPDISK:replica:online,GlueSEUniqueID=grid05.lal.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=6597

# ATLASLOCALGROUPDISK:replica:online, lpnse1.in2p3.fr, GRIF-LPNHE, GRIF, local, grid
dn:GlueSALocalID=ATLASLOCALGROUPDISK:replica:online,GlueSEUniqueID=lpnse1.in2p3.fr,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=1099

# ATLASPRODDISK:replica:online, node12.datagrid.cea.fr, GRIF-IRFU, GRIF, local, grid
dn:GlueSALocalID=ATLASPRODDISK:replica:online,GlueSEUniqueID=node12.datagrid.cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=2199

# cms:replica:online, polgrid4.in2p3.fr, GRIF-LLR, GRIF, local, grid
dn:GlueSALocalID=cms:replica:online,GlueSEUniqueID=polgrid4.in2p3.fr,Mds-Vo-name=GRIF-LLR,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=59977

# ATLASDATADISK:replica:online, grid05.lal.in2p3.fr, GRIF-LAL, GRIF, local, grid
dn:GlueSALocalID=ATLASDATADISK:replica:online,GlueSEUniqueID=grid05.lal.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=6597

# atlas:replica:online, lpnse1.in2p3.fr, GRIF-LPNHE, GRIF, local, grid
dn:GlueSALocalID=atlas:replica:online,GlueSEUniqueID=lpnse1.in2p3.fr,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=3641

# atlasprod:replica:online, node12.datagrid.cea.fr, GRIF-IRFU, GRIF, local, grid
dn:GlueSALocalID=atlasprod:replica:online,GlueSEUniqueID=node12.datagrid.cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=2205

# ATLASPRODDISK:replica:online, lpnse1.in2p3.fr, GRIF-LPNHE, GRIF, local, grid
```



February 2, 2009

```
dn:GlueSALocalID=ATLASPRODDISK:replica:online,GlueSEUniqueID=lpnse1
.in2p3.fr,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=4398
```

```
# ATLASUSERDISK:replica:online, grid05.lal.in2p3.fr, GRIF-LAL,
GRIF, local, grid
dn:GlueSALocalID=ATLASUSERDISK:replica:online,GlueSEUniqueID=grid05
.lal.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=2199
```

```
# ATLASUSERDISK:replica:online, lpnse1.in2p3.fr, GRIF-LPNHE, GRIF,
local, grid
dn:GlueSALocalID=ATLASUSERDISK:replica:online,GlueSEUniqueID=lpnse1
.in2p3.fr,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=3298
```

```
# ATLASLOCALGROUPDISK:replica:online, grid05.lal.in2p3.fr, GRIF-
LAL, GRIF, local, grid
dn:GlueSALocalID=ATLASLOCALGROUPDISK:replica:online,GlueSEUniqueID=
grid05.lal.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=2199
```

```
# ATLASMCDISK:replica:online, lpnse1.in2p3.fr, GRIF-LPNHE, GRIF,
local, grid
dn:GlueSALocalID=ATLASMCDISK:replica:online,GlueSEUniqueID=lpnse1.i
n2p3.fr,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=10995
```

```
# ATLASLOCALGROUPDISK:replica:online, nodel2.datagrid.cea.fr, GRIF-
IRFU, GRIF, local, grid
dn:GlueSALocalID=ATLASLOCALGROUPDISK:replica:online,GlueSEUniqueID=
nodel2.datagrid.cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-
Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=3298
```

```
# permanent:replica:online, grid05.lal.in2p3.fr, GRIF-LAL, GRIF,
local, grid
dn:GlueSALocalID=permanent:replica:online,GlueSEUniqueID=grid05.lal
.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=21468
```

```
# llr:replica:online, polgrid4.in2p3.fr, GRIF-LLR, GRIF, local,
grid
```



February 2, 2009

```
dn:GlueSALocalID=llr:replica:online,GlueSEUniqueID=polgrid4.in2p3.fr,
Mds-Vo-name=GRIF-LLR,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=59977
```

```
# ATLASMCDISK:replica:online, nodel2.datagrid.cea.fr, GRIF-IRFU,
GRIF, local, grid
dn:GlueSALocalID=ATLASMCDISK:replica:online,GlueSEUniqueID=nodel2.d
datagrid.cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=6597
```

```
# atlas:replica:online, nodel2.datagrid.cea.fr, GRIF-IRFU, GRIF,
local, grid
dn:GlueSALocalID=atlas:replica:online,GlueSEUniqueID=nodel2.datagri
d.cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=9705
```

```
# LHCB_DST:replica:online, grid05.lal.in2p3.fr, GRIF-LAL, GRIF,
local, grid
dn:GlueSALocalID=LHCB_DST:replica:online,GlueSEUniqueID=grid05.lal.
in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=1099
```

```
# alice:replica:online, nodel2.datagrid.cea.fr, GRIF-IRFU, GRIF,
local, grid
dn:GlueSALocalID=alice:replica:online,GlueSEUniqueID=nodel2.datagri
d.cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=0
```

```
# ATLASDATADISK:replica:online, nodel2.datagrid.cea.fr, GRIF-IRFU,
GRIF, local, grid
dn:GlueSALocalID=ATLASDATADISK:replica:online,GlueSEUniqueID=nodel2
.datagrid.cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=4398
```

```
# lhcb:replica:online, lpnsel.in2p3.fr, GRIF-LPNHE, GRIF, local,
grid
dn:GlueSALocalID=lhcb:replica:online,GlueSEUniqueID=lpnsel.in2p3.fr
,Mds-Vo-name=GRIF-LPNHE,Mds-Vo-name=GRIF,Mds-Vo-name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=1966
```

```
# ATLASUSERDISK:replica:online, nodel2.datagrid.cea.fr, GRIF-IRFU,
GRIF, local, grid
```



```
dn:GlueSALocalID=ATLASUSERDISK:replica:online,GlueSEUniqueID=node12
.datagrid cea.fr,Mds-Vo-name=GRIF-IRFU,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=2199

# ATLASPRODDISK:replica:online, grid05.lal.in2p3.fr, GRIF-LAL,
GRIF, local, grid
dn:GlueSALocalID=ATLASPRODDISK:replica:online,GlueSEUniqueID=grid05
.lal.in2p3.fr,Mds-Vo-name=GRIF-LAL,Mds-Vo-name=GRIF,Mds-Vo-
name=local,o=grid
GlueSACapability: InstalledOnlineCapacity=6597
```

2.3 Validation procedures

It will be necessary to validate the information published by sites against the reality of the resources they are providing to WLCG.

There are numerous possibilities:

1. GStat: Traditionally GStat has been responsible for checking that required attributes are published. New attributes such as *GlueSACapability: InstalledOnlineCapacity=<size>* and *GlueSACapability: InstalledNearlineCapacity=<size>* can be checked for by GStat. Furthermore, detailed validation scripts are being added to verify the full coherency of the information published by a site.
2. Site managers can look at the WLCG centrally collected results and decide if they believe them to be correct. The NGS Storage Accounting [10] interface is currently the easiest visually to view the collected Installed and Used Storage Capacity.

2.4 Clients queries

In what follows we list the queries performed by the WLCG FTS and lcg-utils/gfal clients in order to select storage resources. Furthermore, the queries and algorithms used by the WLCG WMS are also listed.

1. Get SRM type and endpoint given the SE Unique ID.

- Output attributes: GlueSEPort, GlueSEStatus (not used for the moment), GlueServiceVersion, GlueServiceType, GlueServiceEndpoint

- Request:
* " (| (GlueSEUniqueID=<SRMHOST-FQDN>) (& (GlueServiceType=srm*) (GlueServiceEndpoint=*://<SRM-FQDN>:<PORT>*))) "

2. Get VOInfo object given the SE Unique ID.

- Output attributes: GlueVOInfoPath, GlueChunkKey (for possible GlueSALocalID)

- Request:
* " (& <ACBRs> (GlueVOInfoTag=<spacetokendesc>)



February 2, 2009

```
(GlueChunkKey=GlueSEUniqueID=<SRMHOST-FQDN>)"  
* "(& <ACBRs> (!(GlueVOInfoTag=*))  
(GlueChunkKey=GlueSEUniqueID=<SRMHOST-FQDN>)" (if no spacetoken)
```

3. Get SA object (if no path in VOInfo, or no VOInfo) given SA Local ID and SE Unique ID.

- Output attributes: GlueSAPath, GlueSARoot (for backward compatibility)

- Request:

```
* "(& (GlueSALocalID=<SALocalID_or_VO>  
(GlueChunkKey=GlueSEUniqueID=<SRMHOST-FQDN>))"
```

WMS queries:

4. Find all storage objects that satisfy the following filter:

```
(|(objectclass=gluecesebind)|(objectclass=gluese)|(objectclass=gluesa)  
|(objectclass=glueseaccessprotocol)(objectclass=gluesecontrolprotocol)))
```

Match the user request against the object returned

5. Find all computing objects that satisfy the following filter:

```
(|(objectclass=gluecesebind)(objectclass=gluecluster)(objectclass=gluesubc  
luster)  
(objectclass=gluevoview)(objectclass=gluece))
```

Match the user request against the object returned.

As of today (glite v3.1) the WMS relies on the fact that the SEID is equal to the hostname of the SURL of a specified file (JDL Input/OutputData).



REFERENCES

- [1] Glue Schema Specification version 1.3, Final – 16 Jan 2007,
http://forge.cfnf.infn.it/plugins/scmsvn/viewcvs.php/*checkout*/v_1_3/spec/pdf/GLUESchema.pdf?rev=48&root=glueschema
- [2] Wikipedia definition of CPU: <http://en.wikipedia.org/wiki/CPU>
- [3] Wikipedia definition of Multi-core CPU: <http://en.wikipedia.org/wiki/Multi-core>
- [4] http://goc.grid.sinica.edu.tw/gocwiki/How_does_GStat_count_CPUs
- [5] Storage Element Model for SRM 2.2 and GLUE schema description, F. Donno et al., v. 3.5, 27 October 2006, <https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.glue-wg/docman.root.background.specifications/doc14619;jsessionid=58E33DC10A69FABED90ACD4C8EFE6E1F>
- [6] Recommendations for Changes in gLite Authorization, C.Witzig, 14 April 2008,
<https://edms.cern.ch/document/887174/1>
- [7] GStat: <http://goc.grid.sinica.edu.tw/gstat/>
- [8] CASPUR : <http://www.caspur.it/en/>
- [9] GridMAP: <http://gridmap.cern.ch/gm>
- [10] NGS Storage Accounting: <http://goc02.grid-support.ac.uk/accountingDisplay/index.php?loc=3>
- [11] Transition to a new CPU benchmarking unit for the WLCG:
<http://indico.cern.ch/getFile.py/access?contribId=3&sessionId=0&resId=0&materialId=0&confId=49388>