Table of Contents

Extended Release Notes for WMS 3.4.0

1) Supported platforms

Scientific Linux 5.x x86_64

Scientific Linux 6.x x86_64

2) Major changes

* 2.1 GLUE2 purchasing and match-making

2.1.1) Purchasing

- The WMS core module, responsible for the match-making, has two separate purchaser threads for fetching GLUE1.3 and GLUE2.0 information. These can be independently enabled by configuration (WM section), in this way:

```
    EnableIsmIiGlue13Purchasing  =  <boolean>;
    EnableIsmIiGlue20Purchasing  =  <boolean>;
```

To produce a full ASCII dump of the Information System cache, rendered in classad, execute the following command on the WMS.

```
echo "[ command = \"ism_dump\"; ]"> /var/workload_manager/jobdir/new/req0
```

A GLUE 2.0 dump taken from lcg-bdii.cern.ch has been included in this twiki.

The GLUE 1.3 purchaser is always executed before the GLUE 2.0 one. In this way, when the latter starts fetching entries it will find one of the following cases:

1) the entry was already fetched by the GLUE1.3 purchaser. A merge, G13_Ad.Update(G20_Ad), will be done on the entry with same key that was created by the GLUE 1.3 purchaser (GLUECEUniqueId). The rule to understand if an entry refers to the same resource is the following: IsUndefined(OtherInfo.CREAMCEId) ? GLUE2.Computing.Share.ID : OtherInfo.CREAMCEId The same entry will be selectable by both G13 and G20 requirements. Furthermore, the match-making based on data will keep on working, provided that the JDL refers to G13 attributes.

2) The entry was NOT already purchased by the GLUE1.3 purchaser. A new insertion of the GLUE 2.0 resource will be done. That entry will only be matched by referring to GLUE2.0 attributes in the JDL requirements and rank expressions.

While GLUE1.3 is represented in the WMS as a flattened structure of key/value pairs, mostly, the GLUE2.0 retains the full blown tree of the schema.

Computing Element:

```
[
    info =
        [
            GLUE2 =
                [
                    Computing =
                        [
                            Endpoint =
                                [
                                    Semantics =
                                       {
                                         // this is a classad list of strings
                                       };
                                    Implementor = "string";
                                    WSDL =
                                       {
                                         // this is a classad list of strings
                                       };
                                    ServingState = "string";
                                    Name = "string";
                                    ID = "string";
                                    HealthState = "string";
                                    SupportedProfile =
                                       {
                                         // this is a classad list of strings
                                       };
                                    Staging = "string";
                                    Capability =
                                       {
                                         // this is a classad list of strings
                                       };
                                    QualityLevel = "string";
                                    StartTime = "string";
                                    TrustedCA =
                                       {
                                         // this is a classad list of strings
                                       };
                                    ImplementationName = "string";
                                    DowntimeInfo = "string";
                                    InterfaceName = "string";
                                    OtherInfo =
                                        [
                                            HostDN = "string";
                                            MiddlewareVersion = "string";
                                            MiddlewareName = "string"
                                            // ...
                                        ];
```

```
                 Technology = "string";
                 IssuerCA = "string";
                 Policy =
                    {
                      // this is a classad list of strings (in gLite, elem
                    };
                 URL = "string";
                 HealthStateInfo = "string";
                 ImplementationVersion = "string";
                 InterfaceVersion = "string";
                 JobDescription =
                    {
                      // this is a classad list of strings
                    }
             ];
         Manager =
             [
                 Name = "string";
                 ID = "string";
                 ProductVersion = "string";
                 ProductName = "string"
             ];
         Share =
             [
                 DefaultCPUTime = <integer>;
                 MaxWallTime = <integer>;
                 RunningJobs = <integer>;
                 WaitingJobs = <integer>;
                 ServingState = "string";
                 MaxCPUTime = <integer>;
                 FreeSlots = <integer>;
                 EstimatedAverageWaitingTime = <integer>;
                 ID = "string"; // NB: this will be used to hook up possib
                         InfoProviderVersion = "string"
                 MappingQueue = "string";
                 TotalJobs = <integer>;
                 DefaultWallTime = <integer>;
                 Description = "string";
                 OtherInfo =
                     [
                         InfoProviderName = "string";
                         InfoProviderHost = "string";
                         CREAMCEId = "string"; // NB: this will also be us
                         InfoProviderVersion = "string"
                         // ...
                     ];
                 Policy =
                    {
                      // this is a classad list of strings
                    };
                 EstimatedWorstWaitingTime = <integer>;
                 MaxRunningJobs = <integer>;
             ];
         Service =
             [
                 Name = "string";
                 Type = "string";
                 ID = "string";
                 Capability =
                    {
                       "string"
                    };
                 QualityLevel = "string";
                 Complexity = "string"
             ]
Benchmark =
    {
```

```
                              [
                                  Name = "string";
                                  Type = "string";
                                  ID = "string";
                                  Value = "string";
                              ]
                       };
                    ApplicationEnvironment =
                        [
                            AppName =
                               {
                                  // this is a classad list of strings
                               }
                        ]
                ];
            PurchasedBy = "ism_ii_g2_purchaser"
        ]
]
```

Storage Element:

```
[
    info =
        [
            GLUE2 =
                [
                    Storage =
                        [
                            Endpoint =
                                [
                                    Semantics =
                                       {
                                         // this is a classad list of strings
                                       };
                                    Implementor = "string";
                                    WSDL =
                                       {
                                         // this is a classad list of strings
                                       };
                                    ServingState = "string";
                                    ID = "string";
                                    HealthState = "string";
                                    SupportedProfile =
                                       {
                                         // this is a classad list of strings
                                       };
                                    Capability =
                                       {
                                         // this is a classad list of strings
                                       };
                                    QualityLevel = "string";
                                    TrustedCA =
                                       {
                                         // this is a classad list of strings
                                       };
                                    ImplementationName = "string";
                                    DowntimeAnnounce = "string";
                                    InterfaceName = "string";
                                    InterfaceExtension =
                                       {
                                          ""
                                       };
                                    OtherInfo =
                                        [
                                        ];
                                    Technology = "string";
```

```
                                        IssuerCA = "string";
                                        Policy =
                                           {
                                             // this is a classad list of strings
                                           };
                                        URL = "string";
                                        HealthStateInfo = "string";
                                        ImplementationVersion = "string";
                                        InterfaceVersion = "string";
                                        GLUE2StorageEndpointStorageServiceForeignKey = "string"
                                    ];
                                Manager =
                                    [
                                        ID = "string";
                                        StorageServiceForeignKey = "string";
                                        ProductVersion = "string";
                                        ProductName = "string"
                                    ];
                                Share =
                                    [
                                        Tag = "string";
                                        ServingState = "string";
                                        MaximumLifeTime = <integer>;
                                        ID = "string";
                                        ExpirationMode = "string";
                                        Capacity =
                                            [
                                                TotalSize = <integer>;
                                                Type = "string";
                                                ID = "string";
                                                FreeSize = <integer>;
                                                ReservedSize = <integer>;
                                                UsedSize = <integer>;
                                            ];
                                        StorageServiceForeignKey = "string";
                                        Description = "string";
                                        RetentionPolicy =
                                           {
                                             // this is a classad list of strings
                                           };
                                        DefaultLifeTime = <integer>;
                                        Path = "string";
                                        AccessMode =
                                           {
                                             // this is a classad list of strings
                                           };
                                        AccessLatency = "string";
                                        SharingID = "string"
                                    ]
                            ]
                    ];
                PurchasedBy = "ism_ii_g2_purchaser"
            ]
    ]
```

2.1.2) Match-making with GLUE2.0

- GLUE1.3 and 2.0 attributes are referenced in the JDL through the usual 'other.' operator (see next section), specified in the JDL requirements and rank expressions. Selection on GLUE13 or GLUE20 attributes is simply enabled by referring to the appropriate attributes in the JDL. GLUE20 will be accessed through the structure shown above.

In this release the JDL attribute DataRequirements (and all the deprecated attributes to do match-making with data) will not work on GLUE2.0 resources.

Some examples of match-making:

Match on GLUE13 resources. Same as it has always been:

```
Requirements = other.GlueCEStateWaitingJobs < 100;
Rank = -other.GlueCEStateWaitingJobs;
```

Match on GLUE20 resources:

```
Requirements = other.GLUE2.Computing.Share.WaitingJobs < 100;
Rank = -other.GlueCEStateWaitingJobs;
```

Match on all GLUE13 and GLUE20 resources:

```
wj = iff(isdefined(other.GlueCEStateWaitingJobs), other.GlueCEStateWaitingJobs, other.GLUE2.Compu
Requirements = wj < 100;
Rank = -wj;
```

IMPORTANT: the requirements expressed in the JDL are appended to *WmsRequirements*, a WMS defined classad expression, that includes authZ checks and various constraints on the resource side, automatically appended in && to the user requirements. The present default for *WmsRequirements* only includes GLUE13 attributes. So, according to what type of purchaser is enabled by configuration, the *WmsRequirements* expressions must be adapted accordingly. The default is:

```
 ((ShortDeadlineJob =?= TRUE) ? RegExp(".*sdj$", other.GlueCEUniqueID) : !RegExp(".*sdj$", other.
```

To match only GLUE2 resources, replace with this expression:

```
((member(CertificateSubject, other.GLUE2.Computing.Endpoint.Policy) || member(strcat("VO:", Virtu
```

to match both GLUE 1.3 and GLUE 2.0, just place the two expressions above in or. Shortly:

WmsRequirements = *GLUE13Reqs* || *GLUE2Reqs*;

To test a simple match-making, after having enabled the GLUE 2.0 purchaser in the WM, as described, create a requirements expression like this one in your JDL:

```
requirements = other.GLUE2.Computing.Endpoint.Name == "ce.csl.ee.upatras.gr_org.glite.ce.CREAM";
```

Then, in the WM section of the WMS configuration, comment out the WmsRequirements expression and replace it with an expression referring to valid GLUE2.0 attributes (WmsRequirements=true can be a good start) . Restart the **wmproxy** service. Now submit a list-match request, making sure that no extra default is taken by some inherited configurations.

```
[mcecchi@ui ~]$ glite-wms-job-list-match -a -c glite_wmsui.conf  --endpoint https://devel09.cnaf.

Connecting to the service https://devel09.cnaf.infn.it:7443/glite_wms_wmproxy_server

==========================================================================

           COMPUTING ELEMENT IDs LIST
 The following CE(s) matching your job requirements have been found:

   *CEId*
 - ce.csl.ee.upatras.gr:8443/cream-pbs-dteam
 - ce.csl.ee.upatras.gr:8443/cream-pbs-see

==========================================================================
```

* 2.3 Argus authZ for access control

There are two new variables that need to be specified in the siteinfo file:

```
USE_ARGUS=<boolean>
ARGUS_PEPD_ENDPOINTS="list_of_space_separated_URLs" # i.e.: "https://argus01.lcg.cscs.ch:8154/aut
```

For the moment only openssl oneline old DN format is accepted. This PIP Argus PIP must be enabled in the PEPd configuration. The default authorization mechanism for access control is the gridsite GACL. The WMS automatically sets the Argus resource id to the service endpoint.

* 2.4 Dagmanless DAGs

The WMS used to implement submission and management of DAG jobs via Condor DAGMan. A DAG engine has now been implemented in the WMS engine (WM). This change is to a great extent transparent to the end user.

Nodes are re-evaluated in the WM every MatchRetryPeriod. This design change will introduce several advantages:

1) Simpler, less error prone architecture (integrating DAGMan required several helper modules, limiting mechanisms and the like)

2) Higher performance on DAG jobs, especially those composed by a high number of nodes

3) Ability to handle DAGs with run-time errors so as not to leave the DAG in pending states.

* 2.5 Condor 7.8.0

This change will mainly enable submission to GRAM5 CEs.

NOTES:

- Condor 7.8.0 will install in FHS compliant paths

- Condor 7.8.0 is maintained and repackaged in the EMI WMS through a separate package, named condor-emi. condor-emi does not only provides the official Condor 7.8.0, but also includes several fixes (concerning security and submission to ARC) that have not been ported upstream by the Condor team yet.

* 2.6 Support for RFC proxies (bug #88128)

LCG-CE not expected to support them. OSG, ARC and CREAM CEs will do.

Configuration Changes

- All the configuration parameters used to tune DAGMan (dismissed) are now removed. They were: DagmanLogLevel, DagmanLogRotate, DagmanMaxPre, MaxDAGRunningNodes

- In the wmproxy rpm, several utilities were distributed: glite_wms_wmproxy_purge_proxycache_binary (now replaced by a script), glite_wms_wmproxy_gacladmin (used to generate GACL files) and glite_wms_wmproxy_gridmapfile2gacl, used to generate a GACL file allowing the same entries as speficied in the gridmap file. All these tools have now been removed.

- removed parameter bulkMM (boolean). The bulk match-making feature (group together collections nodes with same requirements, rank) is now aleays enabled.

- The configuration parameters filelist used in WMProxy, WM, ICE, JC and LM sections. The filelist based approach for storing requests is slow and prone to corruption, there's no need to maintain it anymore.

- removed configuration parameter locallogger in wmp (of course the locallogger is the local logger, so there's no need for a parameter, the default (localhost) will be simply taken.

- removed IsmIiLDAPSearchAsync, once used for using asynchronous LDAP primitives for querying the BDII.

- added ice.proxy_renewal_timeout = 120. This parameter was introduced to add a watchdog to possibly hanging renewal requests sent to myproxy. The parameter defines the duration of the grace period.

- added wm.IiGlueLib (= "libglite_wms_ism_ii_purchaser.so.0";)

- added wm.EnableIsmIiGlue20Purchasing and EnableIsmIiGlue13Purchasing, already mentioned in this document

- added attributes to define LDAP filters for GLUE2.0 II purchasers.

wm.IsmIiG2LDAPCEFilterExt allows to specify a custom selection query to populate the Computing Element data structure.

wm.IsmIiG2LDAPSEFilterExt allows to specify a custom selection query to populate the Storege Element data structure.

Defaults are:

IsmIiG2LDAPCEFilterExt =
"(|(&(objectclass=GLUE2ComputingService)(|(GLUE2ServiceType=org.glite.ce.ARC)(GLUE2ServiceType=org.glite

IsmIiG2LDAPSEFilterExt =
"(|(objectclass=GLUE2StorageService)(|(objectclass=GLUE2StorageManager)(|(objectclass=GLUE2StorageShare)(|(c

- added attributes for specifying shared objects to be instantiated by dlopen (helper, purchasers)

- added wm.EnabledReplanner (boolean, **defaults to FALSE**). The job replanner can now be toggled by configuration. The replanning feature is not always used, and in some cases it can show problems with queries to the LB, in case of high load. For this reason it is now disabled by default.

- reduced wm.MatchRetryPeriod 600 -> 300. It now also indicates the time interval with which DAGs are evaluated by the WM engine

- wm.SbRetryDifferentProtocols = true by default in WM conf

- wm.WmsRequirements now has also the queue requirements, that were hard-coded before

3) Known Issues

If a job has support for proxy renewal enabled, an error will be returned when retrieving the output:

Proxy exception: Unable to get Not Before date from Prox"

WORKAROUND: this change will be overwritten in case of a YAIM reconfiguration

add at the end of the file /etc/profile.d/grid-env.sh the line:

gridenv_set "ICE_DISABLE_DEREGISTER" "1"

restart the glite-wms-ice service:

# /etc/init.d/glite-wms-ice restart

- Support for GLUE 2.0 is incomplete for what concerns match-making with the JDL attribute datarequirements.

- ARC CE 12.x is not supported, 11.x is.

- The job perusal feature does not work. Output retrieval fails with error "The Operation is not allowed: the job is not started"

```
[mcecchi@devel15 ~]$ glite-wms-job-perusal --get -f std.out https://devel09.cnaf.infn.it:9000/JJt

Connecting to the service https://devel09.cnaf.infn.it:7443/glite_wms_wmproxy_server


Error - WMProxy Server Error
The Operation is not allowed: the job is not started
Error code: SOAP-ENV:Server
```

WORKAROUND

When getting the above error, you need to rename a file on the WMS:

- The LCMAPS logging verbosity has changed (log file is now var/log/glite/lcmaps.log), so do not expect the long output that was produced before.

4) Code changes

EMI2 WMS introduces several code changes, to comply with the strict EMI JRA1 KPI (Key Performance Indicators) in terms of consolidation and reduction of lines of code. In particular, a non trivial refactoring of authN/Z in wmproxy took place, together with the removal of the dependency on Condor DAGMan, another big change. For the build, Cmake started replacing autotools throughout. Roughly speaking:

~6k lines of code were added in existing modules

```
[mcecchi@devel10 34]$ find . -name emi1_diff|xargs grep -c -e'^[>].*'
./emi.wms.wms-manager/emi1_diff:1114
./emi.wms.wmproxy/emi1_diff:1754
./emi.wms.wms-brokerinfo/emi1_diff:197
./emi.wms.jobsubmission/emi1_diff:768
./emi.wms.wms-helper/emi1_diff:297
./emi.wms.wms-broker/emi1_diff:36
./emi.wms.ice/emi1_diff:323
./emi.wms.wms-matchmaking/emi1_diff:44
./emi.wms.wms-common/emi1_diff:475
./emi.wms.wms-ism/emi1_diff:383
```

plus added another ~1600 (1622) SLOC from new modules (i.e. ISM GLUE2.0 purchasers).

Total added = 6k + 1.5k = 7.5k

~8k lines of legacy code removed

```
[mcecchi@devel10 34]$ find . -name emi1_diff|xargs grep -c -e'^[<].*'
./emi.wms.wms-manager/emi1_diff:831  <-- DAG engine
./emi.wms.wmproxy/emi1_diff:2022 <-- authN/Z refactoring, Argus based authZ
./emi.wms.wms-brokerinfo/emi1_diff:200
./emi.wms.jobsubmission/emi1_diff:1464 <-- code refactoring, cleanup for not handling DAGs anymor
./emi.wms.wms-helper/emi1_diff:211
./emi.wms.wms-broker/emi1_diff:23
./emi.wms.ice/emi1_diff:374
./emi.wms.wms-matchmaking/emi1_diff:56
./emi.wms.wms-common/emi1_diff:553
./emi.wms.wms-ism/emi1_diff:307
```

plus removed several modules in emi.wms.wms-manager (~2300 lines). They were once needed to integrate Condor DAGMan.

```
[mcecchi@devel02 src]$ wc -l dagman_helper.* dagman_utils.h filelist_utils.h glite-process-counte
  974 dagman_helper.cpp
   72 dagman_helper.h
   38 dagman_utils.h
   58 filelist_utils.h
  375 glite-process-counter-args.c
   48 glite-wms-dag-post.sh
   48 glite-wms-dag-post.sh.in
  769 glite-wms-planner.cpp
 2382 total
```

Total removed = 8k + 2k = 10k

5) Changed documentation

The gLite WMS System administrator guide has been updated with the "Configuration Changes" (https://wiki.italiangrid.it/twiki/bin/view/WMS/WMSSystemAdministratorGuide)

The WMPROXY guide has been updated: https://edms.cern.ch/document/674643/1

https://wiki.italiangrid.it/twiki/bin/view/WMS/WebHome (WMPROXY guide link changed) -- MarcoCecchi - 2012-06-22

This topic: WMS > WMS34ExtendedReleaseNotes
Topic revision: r18 - 2013-03-25 - MarcoCecchi

Ideas, requests, problems regarding TWiki? Send feedback