

Table of Contents

| | |
|---------------------------------------|----------|
| ATP - VO TOPOLOGY FEEDS..... | 1 |
| INTRODUCTION..... | 1 |
| EXAMPLE OF XML FEED..... | 1 |
| XML ELEMENTS..... | 1 |
| XML SCHEMA DEFINITION (XSD) FILE..... | 1 |
| VO FEEDS USING LATEST XSD FILE..... | 1 |
| VO GROUP NAMING CONVENTIONS..... | 2 |
| HOW TO VALIDATE AN XML FEED..... | 2 |

ATP - VO TOPOLOGY FEEDS

INTRODUCTION

The ATP VO feeds are XML files used by the LHC VOs to define which are the services that they support (from the list of services defined by GOCDB and RSV). At the same time, these feeds are used to define groups of services from the VO perspective. The groups will be used in the future to query the DB and extract the list of services that correspond to each of them. Groups can be Clouds or Tiers, for instance, or what a VO considers as a Site (a group of services mapped to a specific name).

EXAMPLE OF XML FEED

Check this file as an example of a VO feed: [atp_vo_feed_example.xml](#)

XML ELEMENTS

The different elements defining the XML schema are:

- `<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation=`
- Marks the beginning of the XML file and the XSD file that defines the XML schema
- `<title>ATLAS VO topology for SAM/Nagios</title>`
- Just a title for the VO feed.
- `<description>Groups of services defined by ATLAS VO to be used by the SAM/Nagios monitoring`
- A description of the VO feed.
- `<feed_responsible name="Wile Coyote" dn="/DC=ch/DC=cern/OU=Org Units/OU=Users/CN=wco/CN=49`
- Defines who is the person responsible for the content. The DN (certificate Distinguished attribute).
- `<last_update>2010-03-20T15:25:10Z</last_update>`
- The complete date plus hours and minutes in W3C format when the information was updated
- `<vo>cms</vo>`
- The VO name that defines these groups of services.
- `<atp_site name="Australia-ATLAS" infrast="EGEE">`
- Defines a real site, as it appears defined in GOCDB for services in the EGEE infrastructure, the ones defined in the OSG infrastructure. And for each of these sites we will define the we want to include in one or more VO groups.
- `<service hostname="agh3.atlas.unimelb.edu.au" flavour="SRM" endpoint="srm://agh3.atlas.unimelb.edu.au" type="SRM" base_path="/dpm/atlas.unimelb.edu.au/home/atlas/atlasdatadisk/" name="DATA" />`
- A service is identified by a hostname and a flavour. A flavour is any service type defined in the ATP. For SRM services, you can optionally define one or more spacetokens, with attributes 'base_path' and 'name'.
- `<group name="CA" type="ATLAS_Cloud"/>`
- Finally, for each service defined, we specify to which groups it belongs. A group is defined by attributes: 'name' and 'type'. Under one group 'type' we can define many group 'name's in a hierarchy. This can be helpful to query the ATP and retrieve all the groups (services and groups) under one single 'type' attribute. The groups can also be defined outside the `<service>` element, inside the `<atp_site>` element. This means that all the services defined for that site belong to the groups defined inside the `<atp_site>` element.

XML SCHEMA DEFINITION (XSD) FILE

You can use the XSD file [atp_vo_feed_schema.xsd](#) to validate the syntax of your XML VO feed before being consumed by the ATP.

VO FEEDS USING LATEST XSD FILE

- These are the new VO feeds supporting the XSD file:

| VO Name | URL | XSD COMPLIANT |
|---------|-----|---------------|
|---------|-----|---------------|

| | | |
|-------|---|---|
| ALICE | http://pmendez.web.cern.ch/pmendez/alice/Alice-Voboxes-For-Nagios.xml | 😊 |
| ATLAS | http://atlas-agis-api.cern.ch/request/atp/xml/ | 😊 |
| CMS | http://dashb-cms-job/dashboard/request.py/cmssitemap | 😊 |
| LHCb | http://lhcb-portal-dirac.cern.ch/topology/lhcb_topology.xml | 😊 |

VO GROUP NAMING CONVENTIONS

VOs need to ensure that VO feed information should be in a format acceptable to all. The following conventions have been agreed upon.:

- The group_type names are desired to be in following format for Site, Tier, Federation and Cloud groupings:

```
<VOname>_Site
<VOname>_Tier
<VOname>_Federation
<VOname>_Cloud
```

- The following groupings are obligatory in each group.

```
<group name="Tier-x" type="<VOname>_Tier"/>
<group name="<VO SITENAME>" type="<VOname>_Site"/>
```

where *VOname* refers to the name of the VO and *Tier-x* refers to Tier-0, Tier-1, Tier-2, Tier-3 etc.

- For the _Tier group type, the group names should be **Tier-0, Tier-1, Tier-2, Tier-3**

The groupname for Tier group type is fixed as it is required for generation of Tier-wise reports for all the VOs.

An example of CMS VO grouping is given below:

These 2 fields are obligatory in each group.:

```
<group name="Tier-0" type="CMS_Tier"/>
<group name="<VO SITENAME>" type="CMS_Site"/>
```

These 3 fields are optional, but should keep the naming convention:

```
<group name="<VO CLOUD NAME>" type="CMS_Cloud"/>
<group name="<VO GROUP NAME>" type="CMS_Group"/>
<group name="<VO FEDERATION NAME>" type="CMS_Federation"/>
```

For the other groups, VOs can put whatever they want, e.g.:

```
<group name="T0_CH_CERN" type="All Groups"/>
<group name="T0_CH_CERN" type="Tier1s + Tier0"/>
<group name="T0_CH_CERN" type="Tier2s + Tier1s + Tier0"/>
```

HOW TO VALIDATE AN XML FEED

Using an SLC5 machine, you can use the xmlint command, like in this example:

```
xmllint --noout --schema ./atp_vo_feed_schema.xsd lhcb_topology_example.xml
```

Another way using python could be running the following command:

```
python validate_xml.py
```

For that you need to have the xsd file (attached to this twiki page) and your xml file in the same directory and specify the name of your xml file inside the validate_xml.py file. The validate_xml.py file is also attached to this twiki page (but auto-renamed to .txt for security reasons)

This topic: Main > ATPVOfeeds

Topic revision: r17 - 2016-09-19 - unknown



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)