

Sujet:           **Langage de description des jobs JDL**

Auteur           **Nabil Talhaoui (talhaoui@cnrst.ma)**

Diffusion:       **Groupe Master physique informatique**

Information:     **Les attributs de job sont publiés sur le document DataGrid-01-NOT-0101-0\_6-Note et les attributs de ressources sont publiés sur CG4-TutorialExerciceDescription-v1.0-Testbed**

---

## 1. INTRODUCTION

### Un langage extensible de description des tâches ou jobs pour leurs exécutions en grille

- Utilise les attributs pour la description des jobs,
- Spécifie les caractéristiques d’un job relatif à une application, programme exécutable, données d’entrés, etc....
- Définit les caractéristiques des ressources préférées et pré-requis,
- Basé sur un langage de classes CLASSAD “CLASSified ADvertisement” du Condor

### JDL définit un ensemble d’attributs pour le WMS groupés en 2 catégories:

- Les attributs du job:  
Executable, Arguments, StdInput/StdOutput/StdError, OutputSandbox ...
- Les attributs des ressources:  
MinPhysicalMemory, MinLocalDiskSpace, FreeCPUs, RunningJobs, ...

Les attributs se distinguent par deux types :

Ceux qui sont obligatoires et ceux qui sont optionnels

## 2. SYNTAXE

### Un fichier jdl consiste a un ensemble de lignes d’entrées qui se terminent par ‘;’ chacune

- Ligne d’entrée : <attribut> = <valeur> | < liste de valeurs >;
- Attribut : Une chaîne qui représente le nom de l’attribut
- Valeur : Une chaîne qui représente la valeur de l’attribut
  - Chaîne :           “abc” une double quote pour la chaîne
  - Nombre entier :       1234
  - Nombre réel :       12.52
  - Booléen:           “true”, “false”, expression (voir GLUESchema)
- Liste de valeurs : regroupées par <{ }> et séparées par <,>  
Ex : { “abc”, “bcd”, “def” }
- Commentaires sont précédés par ( # ), pour C/C++ (/\* ceci est un commentaire \*/).

**- Attention: JDL est sensible pour les caractères ‘espace’ ou ‘tabulation’  
- Pas de caractère espace ou tabulation doivent être à la fin de ligne.**

### 3. LES ATTRIBUTS DU JOB

Dans le tableau suivant Table 1 la colonne **M** indique les attributs qui sont obligatoires (mandatory). Valeurs par défauts (sont indiquées avec la colonne **with default**) sont assignées par UI à la base de la spécification du fichier de configuration de UI.

Attribute	M	With default	Meaning
<b>Executable</b>	✓		Executable/command name. The user can specify an executable that lies already on the remote CE. The absolute path, possibly including environment variables, of this file should be specified. The other possibility is to provide a local executable name, which will be staged on the CE. In this case only the file name has to be specified as executable. The absolute path on the local file system executable should be then listed in the InputSandbox attribute expression. It is important to remark that if the job needs for the execution some command line arguments, they have to be specified through the Arguments attribute.
<b>Arguments</b>			This is a string containing all the job command line arguments. E.g. an executable sum that has to be started as: \$ sum N1 N2 -out /tmp/result.out has to be specified as: Executable = "sum"; Arguments = " N1 N2 -out /tmp/result.out";
<b>InputData</b>			A list of: - logical file names and/or - physical file names This attribute refers to data used as input by the job; these data are stored in SEs and published in replica catalogues. Listed names have to be prefixed with "LF:" and "PF:" to indicate that they are respectively: logical file names and physical file names. E.g.: InputData = {"LF:<LFN1>", "PF:<PFN>", "LF:<LFN2>"};
<b>StdInput</b>			Standard input of the job. It can be: - just a file name (staging required) - absolute path (available on the CE) The same mechanism as described for the Executable attribute can be applied.

Attribute	M	With default	Meaning
<b>StdOutput</b>			Standard output of the job. The user has to specify just the file name. To have this file staged back on the submitting machine he/she has to list the file name also in the OutputSandbox attribute expression and use the dg-job-get-output command.
<b>StdError</b>			Standard error of the job. The user has to specify just the file name. To have this file staged back on the submitting machine he/she has to list the file name also in the OutputSandbox attribute expression and use the dg-job-get-output command.
<b>OutputSE</b>			URI of the Storage Element where to store the output data. Once specified, this attribute is used by the RB to choose a CE being “attached” with this SE comparing it with the CloseSE attribute published in the GIS. E.g.: OutputSE = “grid001.cnaf.infn.it”;
<b>InputSandbox</b>			List of files on the UI local disk needed by the job for running. The listed files are staged from the UI to the remote CE. Wildcards and environment variables are admitted in the specification of this attribute. File names can be provided as absolute paths or relative paths starting from the cwd. This attribute is also used to accomplish executable and standard input staging from the submitting machine to the remote execution CE.
<b>OutputSandbox</b>			List of files, generated by the job, which have to be retrieved. The listed files are transferred on the UI local file system by mean of the dg-job-get-output command. Wildcards are admitted in the specification of this attribute. The list shall contain file names (neither absolute nor relative paths).
<b>ReplicaCatalog</b>	✓ (*)		Replica Catalogue Identifier, i.e. something in the following format: <protocol>://<full hostname> :<port>/<Replica Catalog DN>. (* This attribute is mandatory if the InputData attribute has been also specified and contains at least one LFN.

Attribute	M	With default	Meaning
<b>DataAccessProtocol</b>	✓ (*)		This is the protocol or the list of protocols that the application is able to “speak” for accessing InputData on a given SE. The RB matches indeed this attribute with the SEProtocol attribute of published in the GIS. E.g.: DataAccessProtocol = {“file”, “gridftp”}; (* ) This attribute is mandatory if the InputData attribute has been also specified.
<b>Rank</b>	✓	- <b>other.EstimatedTraversa lTime</b>	A ClassAd Floating-Point expression that states how to rank queues that have already met the Constraints expression. Essentially, rank expresses preference. A higher numeric value equals better rank. The RB will give to the job the queue with the highest rank. Default value for this attribute is: -other.EstimatedTraversa lTime
<b>Requirements</b>	✓	<b>TRUE</b>	Boolean ClassAd expression that uses C-like operators. It represents job requirements on resources. In order for a job to run on a given queue, this Requirements expression must evaluate to true on the given queue. Default value for this attribute is TRUE.
<b>Environment</b>			This a list of string representing environment settings that have to be performed on the submitting machine and are needed by the job to run properly. Each item of the list is an equality “VAR_NAME=VAR_VALUE”. E.g.: Environment = {“JOB_LOG=/tmp”, “CNF_PATH=/opt/edg/etc”};

Table 1

## 4. LES ATTRIBUTS DE RESSOURCES

Le tableau suivant Table 1 montres quelques attributs de ressources qui définissent généralement les attributs « Requirements » et « Rank »

CE Attribute	Meaning
<b>CEId</b>	<i>CEId</i> is a string, univocally identifying the CE published in the Grid Information Space. The CEId format is: <full-hostname>:<port-number>/jobmanager-<service>-<queuename> E.g: <i>gtbcg01.ifca.unican.es:2119/jobmanager-pbs-long</i>
	The operating system type and version of the machine or of the

CE Attribute	Meaning
<b>OpSys</b>	machines associated to the queue (assuming that all these machines run the same operating system). E.g.: <i>RH 6.2, SOLARIS 2.6 etc.</i>
<b>FreeCPUs</b>	The total number of free processors associated to the resource, processors able to run, in that moment, jobs submitted to the resource.
<b>TotalJobs</b>	The number of jobs submitted to the resource, jobs that have not already been completed.
<b>RunningJobs</b>	The number of jobs submitted to the resource that are currently running.
<b>IdleJobs</b>	The number of jobs submitted to the resource, jobs that are not running since they are waiting for available resources.
<b>MaxTotalJobs</b>	The maximum number of jobs (running and idle) allowed for the resource.
<b>MaxRunningJobs</b>	The maximum number of running jobs allowed for the resource.
<b>WorstTraversalTime</b>	Worst traversal time (in seconds) for jobs submitted to the Computing Element.
<b>EstimatedTraversalTime</b>	Scaled value of the last traversal time (in seconds), i.e. <i>(Last job traversal time)*(queue length)/(queue length when that job arrived)</i>
<b>Active</b>	This is a boolean attribute indicating if the Computing Element is active. For example if the CE is a queue it indicates if it is ready or not to dispatch jobs to the executing machines.
<b>RunWindow</b>	The time windows that define when the resource is active, (for a queue: ready to dispatch jobs to the executing machines). This attribute may appear zero or more times for a Computing Element entity.
<b>Priority</b>	The priority of the resource.

CE Attribute	Meaning
<b>AFSAvailable</b>	Boolean attribute defining if AFS is installed on the Computing Element.

**Table 2**