

The ARC Job Description Internal Representation

Contents

1 Introduction.....	3
2 Data Structure.....	3
2.1 General Types.....	3
2.1.1 SoftwareVersion.....	3
2.1.2 SoftwareRequirement.....	3
2.1.3 Range.....	3
2.2 JobIdentification.....	3
2.2.1 JobName.....	4
2.2.2 Description.....	4
2.2.3 JobType.....	4
2.2.4 UserTag.....	4
2.2.5 JobVOName.....	4
2.3 Application.....	4
2.3.1 Executable.....	4
2.3.2 Argument.....	5
2.3.3 Input.....	5
2.3.4 Output.....	5
2.3.5 Error.....	5
2.3.6 Join.....	5
2.3.7 Environment.....	5
2.3.8 Prologue.....	5
2.3.9 Epilogue.....	5
2.3.10 LogDir.....	5
2.3.11 RemoteLogging.....	6
2.3.12 Rerun.....	6
2.3.13 SessionLifetime.....	6
2.3.14 ExpiryTime.....	6
2.3.15 ProcessingStartTime.....	6
2.3.16 Notification.....	6
2.3.17 CredentialService.....	6
2.3.18 AccessControl.....	6
2.4 Resources.....	6
2.4.1 OSFamily.....	7
2.4.2 OSName.....	7
2.4.3 OSVersion.....	7
2.4.4 Platform.....	8
2.4.5 NetworkInfo.....	8
2.4.6 IndividualPhysicalMemory.....	8
2.4.7 IndividualVirtualMemory.....	8
2.4.8 IndividualDiskSpace.....	8
2.4.9 IndividualCPUTime.....	8
2.4.10 TotalCPUTime.....	8
2.4.11 IndividualWallTime.....	8
2.4.12 TotalWallTime.....	8
2.4.13 CacheDiskSpace.....	9
2.4.14 SessionDiskSpace.....	9
2.4.15 SessionDirAccessType.....	9

2.4.16 CETYPE.....	9
2.4.17 NodeAccess.....	9
2.4.18 ReferenceTime.....	9
2.4.19 Slots.....	9
2.4.20 CandidateTarget.....	10
2.4.21 RunTimeEnvironment.....	10
2.5 Datastaging.....	10
2.5.1 File.....	10
2.5.2 Directory.....	12
2.5.3 Defaults.....	12
2.6 JobMetaData.....	12
2.6.1 Author.....	12
2.6.2 DocumentExpiration.....	12
2.6.3 Rank.....	12
2.6.4 FuzzyRank.....	12
3 XML Rendering.....	13

1 Introduction

match with GLUE2, description of request, common parts of currently used job description languages, information loss during the transformation take into account. The JobDescription class should be used by all job description handling services in ARC. It should be a common source of job description parsing

CONSISTENT NAMING of Computing Service is required

2 Data Structure

The JobDescription class consist of the following sections: JobIdentification, Application, Resources, DataStaging and JobMetaData. The JobIdentification is used to identify the job using various types, tags and names. The Application class is used to describe explicitly the executable which should be executed at the CE, environment variables, logging, standard in and out, credential service, etc. The Resources class contains information about the type of execution node preferred by the user. The DataStaging class contains information about input file and output files/directories created by the job. Finally the job description may contain meta data, describing the context in which the description was generated, and it is stored in the JobMetaData class.

In the following sections detailed definitions of the classes are given.

2.1 General Types

In this section some complex types which are used generally in different context in the other section are defined.

2.1.1 SoftwareVersion

The SoftwareVersion element is a string specifying the name and version of a specific software which is required by the job. Additionally it MAY contain a comparison operator to specify that the version should be above or below the specified version. The multiplicity is zero or more when contained in a SoftwareRequirements element, otherwise it is zero or one. Skou: Maybe it should be renamed to Software instead. Maybe it should contain two string, one specifying name, and the other version.

2.1.2 SoftwareRequirement

The SoftwareRequirement element specifies the software requirements of a job. It MAY contain multiple other SoftwareRequirement elements, and it MAY contain multiple SoftwareVersion elements. This element MAY contain a boolean that specifies that all OR one of its child elements has to be satisfied. The multiplicity of this element is zero or more when contained in a SoftwareRequirement element otherwise zero or one.

2.1.3 Range

TBD

2.2 JobIdentification

The main goal of this class is to name the job and define its type and identify the job in general.

This identification is used by users in most of the cases. The multiplicity of this element is zero or one.

2.2.1 JobName

This element is a string that MAY be specified by a user to name the job. It may not be unique to a particular job description, which means that a user MAY specify the same `JobName` for multiple job descriptions. Some project defines their own format for the `JobName` in order to categorize and explicitly define the particular version of job they run. However the recommended way to attach user specified categories to the job is to use `UserTag` element instead of stricly formated `JobName`.

If this optional element is not present then it is not defined. The multiplicity of this element is zero or one. The type of this element is `string`.

2.2.2 Description

This element may contain a longer textual description of job. If this optinal element is not present then it is not defined. The multiplicity of this element is zero or one. The type of this element is a `string`.

2.2.3 JobType

This element shows if the job is a member of a greater workflow or just a single job. If this optional element is not present then it is a single job. The multiplicity of this element is zero or one. The type of this element is an `enumeration` with the following elements:

- *collectionelement*: a job submitted as part of a collection of individual jobs which do not communicate among them,
- *parallelelement*: a job submitted as part of a collection of individual jobs which communicate among them,
- *single*: an individual stand-alone job,
- *workflownode*: a job submitted as part of a workflow.

2.2.4 UserTag

This element is for human readable comments, tags for free grouping or identifying different jobs. If this optional element is not present then it is not defined. The multiplicity of this element is zero or more. The type of this element is `string`.

2.2.5 JobVOName

This element should be used to indicate the Virtual Organisation of the job. If this optional element is not present then it is not defined. The multiplcty of this element is zero or one. The type of this element is `string`.

Feri: how can be ensuer that the job really belongs to the given VO? is this related to access control?

2.3 Application

The main goal of this section is to explicitly describe the executable and its runtime environment. The section is **mandatory** and its multiplicity is one.

2.3.1 Executable

This **mandatory** element is a `string` specifying the command to execute. The multiplicity is one.

Feri: tell something about path (relative, absolute etc.)

2.3.2 Argument

This optional element is a `string` specifying an argument for the executable. The multiplicity is zero or more.

Feri: tell something about the order because in case of argument the order may be important

2.3.3 Input

This optional element is a `string` specifying the input (Standard Input) for the Executable. The `Input` element is a filename which should be relative to the session directory of the job. If this element is not present then it is not defined and the consuming system MAY choose any value. The multiplicity is zero or one.

2.3.4 Output

This optional element is a `string` specifying the output (Standard Output) for the Executable. The `Output` element is a filename which should be relative to the session directory of the job. If this element is not present then it is not defined and the consuming system MAY choose any value. The multiplicity is zero or one.

2.3.5 Error

This optional element is a `string` specifying the error output (Standard Error) for the Executable. The `Error` element is a filename which should be relative to the session directory of the job. If this element is not present then it is not defined and the consuming system MAY choose any value. The multiplicity is zero or one.

2.3.6 Join

This is an optional `boolean` element. If it is true it specifies that standard output and error should be joined into the file specified by `Output` element. If it is not defined then the default value is false, and joining is not done. Multiplicity is zero or one.

2.3.7 Environment

This optional element specifies environment variables which should be defined at the CE in the execution environment of the job. It consists of a name/value pair of `strings`. The multiplicity of this element is zero or more.

2.3.8 Prologue

This optional element specifies the path of the executable which will be run before the actual executable is run. If the prologue fails the job execution terminates.

Feri: tell something about path (relative, absolute etc.) and attributes, input, output, error etc variables, use case?

2.3.9 Epilogue

This optional element specifies the path of the executable which will be run after the actual executable is run. If the epilogue fails the job execution terminates.

Feri: tell something about path (relative, absolute etc.) and attributes, input, output, error etc variables, use case?

2.3.10 LogDir

This optional string element defines the name of the directory containing grid-specific diagnostics per job. The defined directory, local to the computing element and relative to the session directory of the job. This directory is kept in the session directory of the job to be available for retrieval. Multiplicity is zero or one.

2.3.11 RemoteLogging

The optional elements specifies an URL for a logging service to send reports about job to. If this element is not defined than the default is set up computing service choose one. It is up to a user to make sure the requested logging service accepts reports from the set of computing service he or she intends to use. Multiplicity is zero or more.

2.3.12 Rerun

An optional integer specifying the number of possible reruns the user can initiate in case of a system failure. If it is not defined, the default value is 0. Multiplicity is zero or one.

2.3.13 SessionLifetime

An optional duration element specifying the maximal time to keep the session directory of the job on the computing service upon job completion. Multiplicity is zero or one.

Feri: default behaviour if not specified should defined

2.3.14 ExpiryTime

The ExpiryTime element is optional and specifies the date and time at which the job is no longer valid. Multiplicity is zero or one.

2.3.15 ProcessingStartTime

This optional date element defines the requested start time of the job. The job execution can be delayed or scheduled using this element. Multiplicity is zero or one.

Feri: default behaviour if not specified should defined

2.3.16 Notification

This optional complex element defines the request for e-mail notifications on job status change. Multiplicity is zero or more.

Feri: structure need to be discussed (open to suggestion) and fits to the generalized state model of the job

2.3.17 CredentialService

This optional URL elements specifies an endpoint which may be used to contact a server to renew/extend delegated proxy credential used by a submitted job. Multiplicity is zero or more.

Feri: default behaviour should defined

2.3.18 AccessControl

Feri: very undefined TBD

2.4 Resources

The mandatory complex resource element describe the resource requirements of the job. Multiplicity is one.

2.4.1 OSFamily

This optional element specifies the general family to which the Execution Environment operating system belongs. Its is open enumeration with the following options which are reflet to GLUE2 OSFamily type:

- linux*: Family of operating systems based on Linux kernel
- macosx*: Family of operating systems based on MacOS X
- solaris*: Family of operating systems based on Solaris
- windows*: Family of operating systems based on Windows
- aix*: AIX
- centos*: CentOS
- debian*: Debian
- fedoracore*: RedHat Fedora
- gentoo*: Gentoo Linux
- leopard*: Mac OS X 10.5 (Leopard)
- linux-rocks*:
- mandrake*: Mandrake
- redhatenterpriseas*: RedHat Enterprise Server
- scientificlinux*: Scientific Linux
- scientificlinuxcern*: Scientific Linux CERN
- suse*: SUSE
- ubuntu*: Ubuntu
- windowsvista*: Microsoft Windows Vista
- windowsxp*: Microsoft Windows XP

Multiplicity is zero or one.

2.4.2 OSName

This optional element is the specific name of the operating system where the job requested to be run. Its is an open enumeration with the following options which are reflet to GLUE2 OSName type:

- aix*: AIX
- centos*: CentOS
- debian*: Debian
- fedoracore*: RedHat Fedora
- gentoo*: Gentoo Linux
- leopard*: Mac OS X 10.5 (Leopard)
- linux-rocks*:
- mandrake*: Mandrake
- redhatenterpriseas*: RedHat Enterprise Server
- scientificlinux*: Scientific Linux
- scientificlinuxcern*: Scientific Linux CERN
- suse*: SUSE
- ubuntu*: Ubuntu
- windowsvista*: Microsoft Windows Vista
- windowsxp*: Microsoft Windows XP

Multiplicity is zero of one.

2.4.3 OSVersion

Optional string element defines the version of the operating system, as defined by the vendor. Multiplicity is zero or one.

Feri: version matchmaking

2.4.4 Platform

Optional element specifies the platform architecture of this Execution Environment. Its is an open enumeration with the following options which are reflet to GLUE2 Platform type:

- amd64*: AMD 64bit architecture
- i386*: Intel 386 architecture
- itanium*: Intel 64-bit architecture
- powerpc*: PowerPC architecture
- sparc*: SPARC architecture

Multiplicity is one or one.

2.4.5 NetworkInfo

This optional element defines the type of internal network connection available among the managed Execution Environment instances. Its is an open enumeration with the following options which are reflet to GLUE2 NetworkInfo type:

- 100megabitethernet*: Network based on 100 MBit/s Ethernet technology
- gigabitethernet*: Network based on 1 GBit/s Ethernet technology
- infiniband*: Network based on Infiniband technology
- myrinet*: Network based Myrinet technology

Multiplicity is zero or one.

2.4.6 IndividualPhysicalMemory

This optional element is a integer range value specifying the amount of physical memory required on each individual computing resource. The amount is given in bytes. Multiplicity is zero or one.

2.4.7 IndividualVirtualMemory

This optional element is a integer range value specifying the required amount of virtual memory for each of the resources to be allocated for this job submission. The amount is given in bytes. Multiplicity is zero of one.

2.4.8 IndividualDiskSpace

This is an optional integer value that describes the required amount of disk space for each resource allocated to the job. The amount of disk space is given in bytes. Multiplicity is zero or one.

2.4.9 IndividualCPUTime

This is optional element is a integer range value specifying the total number of CPU seconds required on each resource to execute the job. Multiplicity is zero or one.

2.4.10 TotalCPUTime

This optional element is a integer range value specifying total number of CPU seconds required, across all CPUs used to execute the job. Multiplicity is zero or one.

2.4.11 IndividualWallTime

This optional element is a integer range that specifies the soft limit on the duration of the

application's execution on each resource, in seconds. Multiplicity is zero or one.

2.4.12 TotalWallTime

This optional element is a integer range that specifies the soft limit on the duration of the application's execution across all CPUs used to execute the job, in seconds. Multiplicity is zero or one.

2.4.13 CacheDiskSpace

This optional integer element contains the requested amount of disk space in bytes on a longer lifetime cache territory. Multiplicity is zero or one.

2.4.14 SessionDiskSpace

This optional integer element contains the requested amount of disk space in bytes on a session directory of the job. Multiplicity is zero or one.

2.4.15 SessionDirAccessType

This optional element defines the type of the remote access to the session directory of the job. It is an enumeration with the following options:

- none: no remote access required
- readable: read only access
- writeable: read and write access

2.4.16 CETYPE

This optional `SoftwareRequirement` element defines which grid middleware the job should be submitted to. Multiplicity is one or more.

2.4.17 NodeAccess

The optional element defines the required connectivity of the computing element where the job should be run. It is an enumeration with the following options:

- inbound: inbound network is required to the running job
- outbound: outbound network is required to the running job
- inoutbound: both direction is required to the running job

Multiplicity is zero or one. If it is not defined then network connection not required for the running job.

2.4.18 ReferenceTime

This optional complex element specifies the value of **what exactly?**

Feri: confusing description, structure need to be defined: TBD

(this requested maximum time for an attribute. This element's value is a requested maximum time for an attribute specified benchmark with an other value specified value on the execution element. A special benchmark is the frequency (default value) where the value can be the CPU Clock.)

2.4.19 Slots

The Resource section have a subsection slots which represents the maximum count of parallel threads or processes.

2.4.19.1 NumberOfProcesses

This optional integer range element specifies the number of instances of the executable that the consuming system **MUST** start when starting this parallel application. If this element is not present then it is not defined and the consuming system **MAY** choose any value. Multiplicity is zero or one.

2.4.19.2 ProcessPerHost

This optional integer range element specifies the number of instances of the executable that the consuming system **MUST** start per host. If this element is not present then it is not defined and the consuming system **MAY** choose any value. Multiplicity is zero or one.

2.4.19.3 ThreadsPerProcesses

This optional integer range element specifies the number of threads per process (i.e., per instance of the executable). If this element is not present then it is not defined and the consuming system **MAY** choose any value. Multiplicity is zero or one.

2.4.19.4 SPMDVariation

This element defines the type of SPMD application.

Feri: enumeration? Which options? Mandatory?

2.4.20 CandidateTarget

This optional complex element represents the computing service preferred to be used. Multiplicity is zero or more.

2.4.20.1 EndPointURL

This mandatory element contains the computing element endpoint's URL. Multiplicity is one.

2.4.20.2 QueueName

This optional string element defines the name of the preferred queue.

2.4.21 RunTimeEnvironment

This optional `SoftwareRequirement` element defines the runtime environment required by the job. Multiplicity is zero or one.

2.5 Datastaging

Data staging defines the files and directories that should be moved to the execution host (stage in) and the files and directories that should be moved from the execution host (stage out). They are staged in before the job starts executing and are staged out after the job terminates (**only if successful?**). If a directory is specified then a recursive copy will be performed if the source or the target have this capability. If the execution environment, source or target does not support recursive copying an error should be reported. The ordering of the elements is not significant.

The `DataStaging` is an optional complex element with multiplicity zero or one. The contains *File*, *Directory* and *Default* sub-elements.

2.5.1 File

This is an optional complex element with multiplicity zero or more.

2.5.1.1 Name

This mandatory string element defines the name of the local file. Multiplicity is one.

Feri: tell something about path: relative, absolute

2.5.1.2 KeepData

This optional boolean element specifies whether the file required to be kept in **where?** After the file staged out. Multiplicity is zero or one. If it is not defined then the default value is false.

2.5.1.3 IsExecutable

This optional boolean element specifies whether the file is executable or not. Multiplicity is zero or one. If it is not defined then the default value is false.

2.5.1.4 DownloadToCache

This optional boolean value shows if the user wants to store the file on the cache disk space. Multiplicity is zero or one. If it is not defined then the default value is false.

2.5.1.5 DataIndexingService

This optional element contains the URL of the data indexing service. Multiplicity is zero or more. If it is not defined then but the Defaults section contains corresponding value than that will be used.

Feri: TBD: how to define different protocols URL may not enough.

2.5.1.6 Source

This optional complex element shows a source (remote location) of the file. There can be more than one instance of this element. If there are multiple sources then one of them will be used. Multiplicity zero or more.

2.5.1.6.1 URL

Remote location of the file. This mandatory element with URL type and multiplicity one.

2.5.1.6.2 Threads

This optional integer element defines how much threads can be used by the given source for staging in the file. If it is not defined the default value is 1. Multiplicity is zero or one.

2.5.1.7 Target

This optional complex element shows a target (remote location) of the file. There can be more than one instance of this element. The goal is uploading the file to at least one target during the stage out phase, but to every mandatory location. Multiplicity is zero or more. From the Source and Target elements at least one should be defined.

2.5.1.7.1 URL

Remote location of the file. This mandatory element with URL type and multiplicity one.

2.5.1.7.2 Threads

This optional integer element defines how much threads can be used by the given source for staging in the file. If it is not defined the default value is 1. Multiplicity is zero or one.

2.5.1.7.3 Mandatory

This optional boolean element defines if the given Target is mandatory or not. Multiplicity is zero or one.

2.5.1.7.4 NeededReplica

This optional integer element can be used to define the required count of the replicas on a given storage system. If it is not defined than the default value is 1. Multiplicity is zero or one.

2.5.2 Directory

Separate element for directory required because the handling at the computing element side of the directory may differs from the files. This is an optional complex element with multiplicity zero or more.

Currently the Directory element have a same subelements what the File element have.

2.5.3 Defaults

This optional complex element contains some default value used during the sage in and stage out process.

2.5.3.1 DataIndexingService

This optional element contains the URL of the data indexing service. Multiplicity is zero or more.

Feri: TBD: how to define different protocols URL may not enough.

2.5.3.2 StagingInBaseURI

This optional URL element defines the endpoint which used for every source file or directory which are defined in a relative way. Multiplicity zero or one.

2.5.3.3 StagingOutBaseURI

This optional URL element defines the endpoint which used for every target file or directory which are defined in a relative way.

2.6 JobMetaData

This is an optional complex element which describe the context in which the job description was generated and provide some further instruction how the specified requirements can be interpreted.

2.6.1 Author

This optional string element may contain the identifier of the exhibitor service. Multiplicity zero or one.

2.6.2 DocumentExpiration

This optional date element contains a timestamp that the job became irrelevant after. It may also be able to handle the different time zones. Multiplicity is zero or one.

2.6.3 Rank

This optional string element states how to rank computing elements that have already met the other requirements. Multiplicity is zero or one.

2.6.4 FuzzyRank

This optional boolean element that enables fuzzyness in the ranking computation. Multiplicity is zero of one. If it is not defined than the default value is false.

3 XML Rendering

Feri: discussion needed espceially related to range support