



*NORDUGRID*

---

NORDUGRID-MEMO-9  
7/5/2005

# **NORDUGRID'S EXTENDED GLOBUS RSL (XRSL) vs JSDL 1.0**

*A COMPARISON*

O. Smirnova, M. Niinimäki



## REMARKS

- Based on
  - JSDL specs draft 0.9.5-03 (May 02, 2005) (<https://forge.gridforum.org/projects/jsdl-wg/document/draft-ggf-jsdl-spec/en/18>)
  - xRSL reference (March 21, 2005) (<http://www.nordugrid.org/documents/xrsl.pdf>)
  - Discussions in EGEE 3<sup>rd</sup> conference, Athens April 2005.
- JSDL elements are outlined according to the nesting suggested in the specification; xRSL elements are not outlined.
- JSDL elements of “complex” type (those having children) are not applicable in xRSL, are marked as “N/A”.
- JSDL substitutes relational operators with range values (realized as XML tags); xRSL uses operators explicitly.
- Allowed multiplicity of elements:
  - if not specified, a single instance is REQUIRED,
  - {0..n} – none is required, many allowed,
  - {0..1} – not more than one allowed,
  - {1..n} – at least one is REQUIRED.
- Clarifications and examples of the use of JSDL Application tag would be most welcome. It should be noticed that the multiplicity of the Application tag is {0..1} and it should encode a lot of information like “chemistry package X in parallel execution environment Y”.
- JSDL “POSIXApplication” is considered an extension in the JSDL specs; it is the best match for an xRSL application.
- The role of the xRSL runTimeEnvironment (RTE in what follows) and its correspondence to JSDL elements have been extensively discussed. Contrary to a widespread opinion, RTE is not another kind of an application: it is an attribute of an application, thus being situated on another level. RTE requirement to a POSIX application is the same as a memory limit requirement: an application must not be scheduled to a resource that does not meet these requirements.
- JSDL “DataStaging” section is still underdeveloped and not very satisfactory defined; difficult to compare/evaluate.
- It is very important to be able to specify alternative requirements like “operating system LINUX OR operating system Solaris required for the job”. From the present JSDL draft it is not clear whether it is possible.
- “JobRefinement” and “Profile” elements could perhaps be used to express alternative requirements as above, however, they are absent from the latest JSDL draft.
- A very important feature of xRSL: possibility to concatenate many job definitions in one request. The present JSDL draft does not describe such possibility.
- Conclusion: core elements (executable, arguments, stdin/stdout/stderr) are present in both structures, can be easily converted. If JSDL is capable of defining several job requests and can be extended by including the missing elements (which seems to be trivial), NorduGrid/ARC will benefit from switching from xRSL to JSDL.

JSDL	xRSL
<b>NUMERICAL OPERATORS</b>	
Not supported, can be used by XML	Not supported
<b>STRUCTURE</b>	
<b>DOCUMENT LAYOUT</b>	
Structured layout with mandatory and optional blocks: <pre>&lt;JobDefinition&gt;   &lt;JobDescription&gt;     &lt;JobIdentification ... /&gt;?     &lt;Application ... /&gt;     &lt;Resource ... /&gt;*     &lt;DataStaging ... /&gt;*   &lt;/JobDescription&gt; &lt;/JobDefinition&gt;</pre>	No structure, flat unordered layout. If an element appears several times, the later values overwrite the earlier ones.
<b>ELEMENT TYPES</b>	
Data types: <ul style="list-style-type: none"> <li>• xsd:string</li> <li>• xsd:normalizedString,</li> <li>• xsd:positiveInteger</li> <li>• xsd:boolean</li> <li>• xsd:NCName</li> <li>• xsd:anyType</li> <li>• xsd:token</li> <li>• xsd:any##other</li> <li>• xsd:complex</li> <li>• jsdl:rangeValue</li> </ul>	Data type not explicitly enforced, de-facto used: <ul style="list-style-type: none"> <li>• string</li> <li>• URL</li> <li>• positiveInteger</li> <li>• time</li> </ul>
Processor architectures: <b>MUST</b> be supported <ul style="list-style-type: none"> <li>• sparc</li> <li>• powerpc</li> <li>• x86</li> <li>• parisc</li> <li>• mips</li> <li>• ia64</li> <li>• arm</li> <li>• other</li> </ul>	N/A
Filesystem types: <b>MUST</b> be supported <ul style="list-style-type: none"> <li>• swap (memory paging)</li> </ul>	N/A

<ul style="list-style-type: none"> <li>• tmp (not available after job end)</li> <li>• spool (may persist)</li> <li>• normal (deleted only by user)</li> </ul>	
Operating System Types: may be supported, see CIM_OperatingSystem model <ul style="list-style-type: none"> <li>• large enumeration (currently 53 items)</li> </ul>	N/A
File Creation Flags: <ul style="list-style-type: none"> <li>• Overwrite</li> <li>• DontOverwrite</li> <li>• Append</li> <li>• Prepend</li> </ul>	N/A
Range Value: a complex type, allows to define open- and/or close-ended ranges, as well as exact values; <b>MUST</b> be supported	N/A
<b>ELEMENT SET</b>	
<b>JOB STRUCTURE</b>	
• JobDefinition	& (ampersand) {1..n}
- JobDescription	N/A
- Description {0..1}	-
- -	rsl_substitution {0..n}
<b>JOB IDENTITY</b>	
• JobIdentification {0..1}	N/A
- JobName {0..1}	jobName {0..n}
- JobAnnotation {0..1}	-
- JobProject {0..n}	-
<b>APPLICATION REQUIREMENTS</b>	
• Application {0..1}	N/A
- ApplicationName {0..1}	
- ApplicationVersion {0..1}	
- Description {0..1}	
• POSIXApplication {0..1} - <b>extension</b>	N/A
- Executable {0..1}	executable {1..n}
- Argument {0..n}	arguments {0..n}
- Input {0..1}	stdin {0..n}
- Output {0..1}	stdout {0..n}
- Error {0..1}	stderr {0..n}
- -	join {0..n}
- -	gmLog {0..n}
- WorkingDirectory {0..1}	-
- Environment {0..n}	environment {0..n}
- TileSize {0..1}	-
- ProcessCount {0..1}	-
- WallTimeLimit {0..1}	-
- FileSizeLimit {0..1}	-
- CoreDumpLimit {0..1}	-
- DataSegmentLimit {0..1}	-
- LockedMemoryLimit {0..1}	-

- MemoryLimit {0..1}	memory {0..n}
- OpenDescriptorsLimit {0..1}	-
- PipeSizeLimit {0..1}	-
- StackSizeLimit {0..1}	-
- CPUTimeLimit {0..1}	cpuTime {0..n}
- ProcessCountLimit {0..1}	-
- VirtualMemoryLimit {0..1}	-
- ThreadCountLimit {0..1}	-
- UserName {0..1}	-
- GroupName {0..1}	-
- -	runTimeEnvironment {0..n}
- -	executables {0..n}
- -	startTime {0..n}
- -	lifeTime {0..n}
- -	notify {0..n}
- -	rerun {0..n}
- -	dryRun {0..n}
- -	jobReport {0..n}
<b>RESOURCE REQUIREMENTS</b>	
• Resource {0..1}	N/A
- CandidateHosts {0..1}	N/A
▪ HostName {1..n}	cluster {0..n}
- -	queue {0..n}
- CPUArchitecture {0..1}	architecture {0..n}
- CPUSpeed {0..1}	-
- CPUTime {0..1}	cpuTime {0..n}
-	gridTime {0..n}
- CPUCount {0..1}	count {0..n}
- ResourceCount {0..1}	-
- PhysicalMemory {0..1}	memory {0..n}
- VirtualMemory {0..1}	-
- NetworkBandwidth {0..1}	-
- -	nodeAccess {0:n}
- DiskSpace {0..1}	disk {0..n}
- FileSystem {0..n}	N/A
▪ MountPoint {0..1}	-
▪ MountSource {0..1}	-
▪ DiskSpace {0..1}	disk {0..n}
▪ FileSystemType {0..1}	-
▪ Description {0..1}	-
- ExclusiveExecution {0..1}	-
- OperatingSystem {0..1}	N/A
▪ OperatingSystemType {0..1}	-
▪ OperatingSystemVersion {0..1}	-
▪ Description {0..1}	-
- -	middleware {0..n}
<b>DATA STAGING</b>	
• DataStaging {0..n}	inputFiles {0..n} outputFiles {0..n}

- FileName	-
- FileSystemName {0..1}	-
- CreationFlag	-
- DeleteOnTermination {0..1}	-
- -	ftpThreads {0..n}
- -	replicaCollection {0..n}
- Source {0..1}	inputFiles/outputFiles source
▪ URI	inputFiles/outputFiles source
▪ -	cache {0..n}
- Target {0..1}	inputFiles/outputFiles target
▪ URL	inputFiles/outputFiles target