

**OpenDSP** is an open architecture implementation of SOAP Web Service multi-user access and policy-based job control using DRMAA (Distributed Resource Management Application API) routines supported by various distributed resource management systems.

**For more details check out: <http://sourceforge.net/projects/opensdp>**

### **JSDL in OpenDSP v0.1**

OpenDSP uses subset of JSDL 1.0 POSIX for job description. The following attributes are currently supported, their meaning can be found in JSDL specification.

- <jSDL:JobName>, in <jSDL:JobIdentification>  
  
Job name.
- <jSDL-posix:POSIXApplication>, in <jSDL:Application>
  - <jSDL-posix:Executable>  
  
Absolute path to executed program.
  - <jSDL-posix:WorkingDirectory>  
  
Working directory for job execution. Path can be absolute or relative to user home directory. If it's not specified then home directory is used.
  - <jSDL-posix:Input>  
  
Specifies file used as job standard input. If not specified job is started with an empty input stream. Path is relative to working directory. The \$drmaa\_incr\_ph\$ placeholder may be used at any position within the path of parametric job and SHALL be substituted by the underlying DRM system with the parametric jobs' index.
  - <jSDL-posix:Output>  
  
Specifies file to which standard output is directed. If not specified it is impossible to retrieve job's standard output using DSP. The \$drmaa\_incr\_ph\$ placeholder may be used at any position within the path of parametric job and SHALL be substituted by the underlying DRM system with the parametric jobs' index.
  - <jSDL-posix>Error>  
  
Specifies file to which standard error is directed. If not specified it is impossible to retrieve job's standard error using DSP. The \$drmaa\_incr\_ph\$ placeholder may be used at any position within the path of parametric job and SHALL be substituted by the underlying DRM system with the parametric job's index.
  - <jSDL-posix:Argument>  
  
This element is a constrained normalized string specifying an argument element for the application. Argument elements can be empty and MUST NOT be collapsed.
- <jSDL:DataStaging>
  - <jSDL:FileName>

Path relative to working directory of the file that will be staged in or out. The `$drmaa_incr_ph$` placeholder may be used at any position within the path of parametric job and will be substituted with the parametric jobs' index.

- `<jsdl:Target>`

A Target element contains the path of the file on the local system (DSP client's). This file will be staged out and saved by the client to the path specified after the job has terminated.

- `<jsdl:Source>`

A Source element contains the path of the file on the local system (DSP client's). This file will be staged in from the location specified before the job has started.

- `<jsdl:CreationFlag>`

This flag specifies whether server will or will not overwrite files when staging in. Possible values are: `overwrite` and `dontOverwrite`. Append flag is not currently supported.

- `<jsdl>DeleteOnTermination>`

This boolean flag specifies whether DSP will remove the file after job terminates (normally or not). If the file is to be staged out (Target element is used), then it will be deleted only after successful stage out. File to be deleted does not have to be connected with current job. Files are guaranteed to be deleted from submit host. Deleting files from execution hosts is only possible if filesystems are shared.

Example job description used by OpenDSP v0.1 and SGE v6u7.

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:JobDefinition
  xmlns:jsdl="http://schemas.ggf.org/jsdl/2005/11/jsdl"
  xmlns:jsdl-posix="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix">
<jsdl:JobDescription>
  <jsdl:JobIdentification>
    <jsdl:JobName>checksum example job</jsdl:JobName>
  </jsdl:JobIdentification>
  <jsdl:Application>
    <jsdl:ApplicationName>md5sum</jsdl:ApplicationName>
    <jsdl-posix:POSIXApplication>
      <jsdl-posix:Executable>/usr/bin/md5sum</jsdl-posix:Executable>
      <jsdl-posix:Argument>-</jsdl-posix:Argument>
      <jsdl-posix:Argument>/non_existent_file_to_generate_stderr</jsdl-posix:Argument>
      <jsdl-posix:Input>infile</jsdl-posix:Input>
      <jsdl-posix:Output>checksum</jsdl-posix:Output>
      <jsdl-posix:Error>stderr</jsdl-posix:Error>
    </jsdl-posix:POSIXApplication>
  </jsdl:Application>
  <jsdl:DataStaging>
    <jsdl:FileName>infile</jsdl:FileName>
    <jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
    <jsdl>DeleteOnTermination>true</jsdl>DeleteOnTermination>
    <jsdl:Source>
      <jsdl:URI>/bin/sh</jsdl:URI>
    </jsdl:Source>
  </jsdl:DataStaging>
  <jsdl:DataStaging>
    <jsdl:FileName>checksum</jsdl:FileName>
    <jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
    <jsdl>DeleteOnTermination>true</jsdl>DeleteOnTermination>
    <jsdl:Target>
      <jsdl:URI>/home/plichocki/md5sum.dsp</jsdl:URI>
    </jsdl:Target>
  </jsdl:DataStaging>
  <jsdl:DataStaging>
    <jsdl:FileName>stderr</jsdl:FileName>
    <jsdl:CreationFlag>overwrite</jsdl:CreationFlag>
    <jsdl>DeleteOnTermination>true</jsdl>DeleteOnTermination>
    <jsdl:Target>
      <jsdl:URI>/home/plichocki/md5sum.err</jsdl:URI>
    </jsdl:Target>
  </jsdl:DataStaging>
</jsdl:JobDescription>
```

</jsdl:JobDefinition>

In OpenDSP v 0.1 data staging is achieved using DIME attachments in SOAP messages, so OpenDSP client must additionally support this feature. Attachments carry path specified in <jsdl:Target> element in "Id" and this can be used by OpenDSP client to save job results in appropriate place.