BES 1.1 Extensions

Status of this Memo

This memo provides information to the Grid community regarding a set of profiles and extensions on the Job Submission Description Language and the OGSA™ Basic Execution Services documents. These extensions are strictly upward compatible, i.e., they are entirely consistent with the existing specifications and are elaborations of XSDany or new porttypes. Distribution is unlimited.

Copyright Notice

Copyright © Open Grid Forum (2013). All Rights Reserved.

Abstract

Contents

[Abstract 1](#_Toc374280761)

[1 Introduction 3](#_Toc374280762)

[2 Notational Conventions 3](#_Toc374280763)

[3 BES State Model Changes 4](#_Toc374280764)

[4 JSDL Profiled Elements 6](#_Toc374280765)

[4.1 Unique Name Attribute 6](#_Toc374280766)

[4.2 Modules 6](#_Toc374280767)

[4.3 Grid FileSystem 6](#_Toc374280768)

[4.4 WallclockTime Restriction 6](#_Toc374280769)

[4.5 Matching Parameters 6](#_Toc374280770)

[4.6 File Staging Creation Enumeration – extension 7](#_Toc374280771)

[4.7 Additional Operating System Types 7](#_Toc374280772)

[4.8 SPMD Extensions 8](#_Toc374280773)

[4.9 Pre and Post Processing 8](#_Toc374280774)

[5 Extensions and Profiles 8](#_Toc374280775)

[5.1 Profiled BES Factory Attributes 8](#_Toc374280776)

[5.1.1 Matching parameters 8](#_Toc374280777)

[5.2 Wallclock 8](#_Toc374280778)

[5.3 File systems 8](#_Toc374280779)

[6 BES porttype extensiions 10](#_Toc374280780)

[6.1.1 SupportBESMatching1.1 10](#_Toc374280781)

[6.1.2 SupportBES1States1.1 10](#_Toc374280782)

[6.1.3 SupportBESGLUE2.0 10](#_Toc374280783)

[6.2 Resource Properties 10](#_Toc374280784)

[6.3 PortTypes 10](#_Toc374280785)

[6.3.1 Resume 10](#_Toc374280786)

[7 Compliance Targets 11](#_Toc374280787)

[8 Security Considerations 11](#_Toc374280788)

[9 Author Information 11](#_Toc374280789)

[10 Contributors 11](#_Toc374280790)

[11 Acknowledgements 11](#_Toc374280791)

[Full Copyright Notice 11](#_Toc374280792)

[Intellectual Property Statement 12](#_Toc374280793)

[Normative References 12](#_Toc374280794)

# Introduction

The Production Grid Interoperability Working Group identified a number of execution management use cases and requirements in GFD.180. A number of ways to meet these requirements have been extensively discussed. They fall into two categories: 1) define a new set of specifications from scratch to meet the requirements, and 2), profile and minimally extend existing specifications to meet the requirements.

The BDP is a part of the second approach, profiling and extending existing specifications to meet the requirement. It combines, extends, and profiles five existing specifications to meet the PGI requirements: WS Addressing EndPoint References, OGSA Basic Execution Services (OGSA\_BES, or BES) [GFD.108], RNS 1.1 OGSA-WSRF Basic Profile 1.0 [GFD.172], WS-Iterator 1.0 [GFD.188], and OGSA-ByteIO WSRF Basic Profile 1.0 [GFD.98].

# Notational Conventions

The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” are to be interpreted as described in RFC-2119 [RFC 2119].

The document refers to a “BES Directory Profile compliant system” as a “Compliant system”.

This specification uses namespace prefixes throughout; they are listed in Table 2‑1. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table ‑: Prefixes and namespaces used in this specification.

|  |  |
| --- | --- |
| Prefix | Namespace |
| xsd | <http://www.w3.org/2001/XMLSchema> |
| wsa | <http://www.w3.org/2005/03/addressing> |
| Rns | <http://schemas.ogf.org/rns/2009/12/rns> |
| Byteio | <http://schemas.ggf.org/byteio/2005/10/byte-io> |
| sbyteio | <http://schemas.ggf.org/byteio/2005/10/streamable-access> |
| rbyteio | <http://schemas.ggf.org/byteio/2005/10/random-access> |
| Bdp | <http://schemas.ogf.org/bdp/2012/03/bdp> |

For all QName aliases used in this document are defined as follows

|  |  |
| --- | --- |
| Alias | Namespace |
| jsdl | http://schemas.ggf.org/jsdl/2005/11/jsdl |
| BESExtensions | http://schemas.ggf.org/BES/2014/BESExtensions |
| gii-common | http://vcgr.cs.virginia.edu/genii/common |
| gii-bes | http://vcgr.cs.virginia.edu/genii/2008/03/bes |

# BES State Model Changes

The BES state model from the original specification contains five states. The BES specification allows the profiling of substates within the five states as long as the state transitions at the top level are not modified.

Pending

Running

Finished

Terminated

Failed

TerminateActivity request

System error/failure event

Successful termination of activity

**Figure 1**. Basic state model from GFD 108.

During a series of meetings within the PGI working group additional requirements were determined, in particular the ability to suspend an activity before and after execution in order to allow external client-driven interaction with the session directory of the job. This was codified in the European Middleware Initiative interfaces [cite]. In addition, there has been significant feedback from developers on the need to explicitly model pre and post processing steps. This profile addresses the lessons learned by defining a new set of sub-states known as the BES-PGI state model. The BES-PGI defines sub states for the Pending and Running states.

The initial BES state model has no transition from Pending to Failed. Experience has shown that such a transition is needed. Therefore, compliant implementations MAY transition from Pending to Failed.





# JSDL Profiled Elements

## Extensions of existing definitions to update

### Operating System

9.3.5.1, from EMI execution service

ARM

### CPU Type

9.3.5.2 and co-processor 9.3.5.2.1 from EMI, general GP-GPU, e.g., AMD opencl, and intel phi

### GFFS FileSystem

We profile another file system type for JSDL. Its’ name is **GFFS** and it indicates that a job wishes to have the GFFS grid namespace mounted for it before it starts running. In all other respects it is just like any other file system in the JSDL specification.

### iROD

### HDFS

### Block Storage File System – S3

## Network Info

See section 9.3.5.3

## RemoteSessionAccess

9.3.5.8

## CoreRequirement

9.3.5.9

## QueueName

9.3.5.10

## Walltime

9.3.5.13

## IndividualCPUTime

9.3.5.11

## ParallelEnvironment

9.3.5.16

## Hold

Question is which element does this go in. In the JSDL or in the createActivity operation. If in the JSDL, where? It does not really fit in identification, application, staging, or resources.

< BESExtensions:Hold>

<PENDING> <Staging-in /> </PENDING>

</ BESExtensions :JSDLModuleLoad> \*

The semantics are that the activity will move into the Pending:Held state AFTER exiting the specified state. Upon resumption the activity will move into the subsequent state in the state model.

## Matching Parameters – Rename to “options” keep in resources

The ability to match jobs to resources is fundamental to Grids. This requirement goes both ways – jobs must be able to specify what they need in a BES, and the BES must be able to specify attributes the job must possess. This can be accomplished with *Matching Parameters*. A Matching Parameter is an arbitrary name/value pair that a job can use to indicate a property that it requires or supports (required properties have a value of **requires:**value and supported properties have a value of **supports:**value). BESs likewise can advertise matching parameters that it supports or requires and they are matched to these JSDL matching parameters in the obvious way. For this JSDL extension, the element is defined as

< PGIExtensions:property name=” *xsd:string* “ value=” *xsd:string* “/> **\***

and it occurs 0 or more times inside of

/ PGIExtensions:Resources

These matching parameters are different from the existing mechanisms built into the JSDL specification in that they allow for a user to make requests for scheduling or matching based on completely arbitrary scheduling parameters that are agreed on outside of the JSDL specification. In theory, this could also have been done by adding new XML elements into the resources section of the JSDL, but by doing it this way, the code can be written to do the simple matching without knowing what the possible space of parameter types and values is ahead of time.

Jobs that **require** a value get matched against BESs that **support** or **require** it, and jobs that **support** a value are allowed to run on BESs that **require** it. The full table of matching is given below. This table shows the ways that Genesis II evaluates whether or not a given job is allowed to be scheduled on a given BES.

|  |  |  |  |
| --- | --- | --- | --- |
|  | BES Supports Parameter | BES Requires Parameter | BES Does Not Advertize Parameter |
| Job Supports Parameter | Allowed | Allowed | Allowed |
| Job Requires Parameter | Allowed | Allowed | Not Allowed |
| Job Does Not Mention Parameter | Allowed | Not Allowed | Allowed |

## File Staging

### Creation Enumeration – extension

JSDL defines *JSDL:CreationFlagEnumeration* flags of overwrite, dontOverwrite, and append.

We define two additional members of the enumeration,

<JSDL:setExecutable boolean>

– if appropriate for the local operating system set the executable bit for the file.

## UMASK and inherited file mask –to do - Shahbaz

### Recursion

<source:URI>

<JSDL:RecursiveCopy>

Semantics would be IF and only IF a) the URI protocol supports recursive operations AND the middleware supports recursive operations, copy recursively. Else fault.

### Wild Cards

1. Wild card will be expressed in the URI element.
2. Wild card expansion is the responsibility of the middleware (the BES), if the protocol supports wild cards, and the middle wants to use it, that is ok. Conversely, if the protocol does not support wildcards, the middleware MUST or fault.
3. File & directory structure will be the same.
4. At a minimum the middleware should support “\*” and “?” What escape character if any?

The middleware is free to perform optimizations as possible, e.g., use tar and ssh instead of scp

## Pre and Post Processing

Often it is desirable to run a script before or after executing an application. For a sequential application this does not represent a problem – a sequential combination of the preprocessing steps, the application, and the post-processing steps can be combined into a single script. When the application is to be run in a parallel environment, e.g., an MPI application, one cannot simply run the preprocessing on every node. Instead it is desired to run the preprocessing steps on the head or login node, execute the job on the parallel machine (usually through a batch scheduler) and then execute the post processing steps.

To support this use case we profile the JSDL extension to add an optional element to the Application element. There may be AT MOST ONE of pre and post process.

This will be a sub-element of JSDL:Application

<BESExtensions:ApplicationPreExecution>

Path arguments what-do-do-if-fails

</ BESExtensions:ApplicationPreExecution > \*

COPY EXECUTABLE TYPE FROM EMI-ES

<BESExtensions:ApplicationPostExecution>

myPostProcessingScript

</ BESExtensions:ApplicationPostExecution > \*

Note that the order that these appear in the application section is irrelevant.

# Extensions and Profiles

## Profiled BES Factory Attributes

For each of the JSDL things above we need the corresponding thing in the BES factory attributes and/or the Resource Properties.

GLUE 2 computeelements

# BES porttype extensions

This section describes the compliance requirements.

### SupportBESMatching1.1

This Metadata entry is OPTIONAL. If present in the Endpoint Reference for the BES it whether the endpoint is compliant with the BES this profile Profile. The entry’s type is xsd:boolean, and it has a cardinality of exactly 1. A value of “true” indicates compliance, while a value of “false” or the absence of the entry indicates non-compliance.

<wsa:EndpointReference>

...

<wsa:Metadata>

<bdp:whatever>

/xsd:boolean

</bdp:SupportsBESDirectory>

</wsa:Metadata>

</wsa:EndpointReference>

### SupportBES1States1.1

### SupportBESGLUE2.0

## PortTypes

### TODO: Add optional WS-Notification subscription in create activity,

This section is applicable to implementations that implement the WSRF-BP.

Users may want to subscribe to state change and resource exceeded events.

WSRF-BP, section 6.1 mandates the use of WSNotification’s NotificationProducer port-type, which enables a consumer to subscribe to notification topics at any time via the Subscribe message exchange. The topics that are available from the port-type are not mandated, however, so the following topics are OPTIONAL.

Comment2: How do we subscribe within the bes framework? May be through the createactivity method similar to the way genesis does?

**Topic: State Change**

Notification returns EPR of the activity, old state and new state, and some (optional) description

Topic:

/aep:ActivityStateChanged

/aep:ActivityStateChanged/aep:ReachedFinalState

Returns:

<aep:BESActivityStateChangedContents>

<aep:ActivityEPR>/wsa:EndpointReferenceType</aep:ActivityEPR>

<aep:OldState>bes:ActivityStatus</aep:OldState>

<aep:NewState>bes:ActivityStatus</aep:NewState>

<aep:Description>xsd:string</aep:Description> ?

</aep:BESActivityStateChangedContents>

The notification response message is returned to WSNotification’s Consumer service for which the endpoint address is given at the time of subscription. The Consumer service could be residing at the user facing client side entity, meta-scheduler, or portal application.

### ResumeActivities

This operation requests that a specified set of activities be released from either Pending:HOLD-IN or Running:HOLD-Out state. The BES transitions the specified activities from either the Pending:HOLD-IN state to Pending:Stage-In or the Running:HOLD-OUT state to Running:Stage-Out. If an activity cannot be released immediately, the eventual success of this operation (i.e., to move the activity into the Running:*Hold-OUT* state) must be determined through other operations (e.g., GetActivityState) or by subscribing to any generated events, if a BES supports subscription.

If a request is successful, then each specified activity will eventually enter the corresponding staging state. Invoking this operation on an activity not in a HELD state has no effect.

#### Input(s)

* **EPR[] activities**: A vector of zero or more EPRs identifying the activities that are to be resumed.

#### Output(s)

* **ResumeActivityResponseType[] Response**: A vector detailing the responses of the BES to the Resume requests. The *Resume* element is a boolean value indicating whether the BES successfully (true) resumed the activity or not (false). If true is returned, then the associated activity has now exited the HELD state. If false is returned then the activity MAY eventually transition out of the *HELD* state. If an activity specified in the input cannot be located or cannot be resumed for some reason, or is not in a HELD state, then the ResumeResponse MUST contain a SOAP-1.1 fault element instead of a Resumed element.

<ResumeActivityResponse>

<ActivityIdentifier>EPR</ActivityIdentifier>

<Resumed> xsd:Boolean </Resumed> ?

|

<SOAP-1.1:fault> ... </SOAP-1.1:fault> ?

</ResumeActivityResponse> \*

#### Fault(s)

* **InvalidRequestMessageFault:** An element in the request message is not recognized. The elements that are not recognized are described in the body of the fault. This does not mean that the element itself is in error, but rather that it specifies a syntactically correct value which does not in fact make sense.

# Compliance Targets

# Security Considerations

Access control is out of scope.

# Author Information

Daniel Dougherty

University of Virginia

Andrew Grimshaw (editor)  
University of Virginia

Shahbaz Memnon  
Forschungszentrum Juelich (FZJ)

Bernd Schuller  
Forschungszentrum Juelich (FZJ)

# Contributors

We gratefully acknowledge the contributions made to this specification by [insert names].

# Acknowledgements

We are grateful to numerous colleagues for discussions on the topics covered in this document, in particular (in alphabetical order, with apologies to anybody we've missed)[insert names].

We would like to thank the people who took the time to read and comment on earlier drafts. Their comments were valuable in helping us improve thereadability and accuracy of this document.

Full Copyright Notice

Copyright © Open Grid Forum (2014). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE OPEN GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director (see contact information at OGF website).

Normative References

[RFC 2119] Bradner, S. Key words for use in RFCs to Indicate Requirement Levels. Internet Engineering Task Force, RFC 2119, March 1997.Available at <http://www.ietf.org/rfc/rfc2119.txt>

[JSDL10] Available at <http://www.ggf.org/documents/GFD.136.pdf>

OGF

WS Addressing EndPoint References

OGSA Basic Execution Services (OGSA\_BES, or BES) [GFD.108]

RNS 1.1 OGSA-WSRF Basic Profile 1.0 [GFD.172]

WS-Iterator 1.0 [GFD.188]

OGSA-ByteIO WSRF Basic Profile 1.0 [GFD.98]