Activity Endpoint Profile: V.0.1

Status of this Memo

This memo provides information to the Grid community regarding the specification of the Activity Endpoint Profile:

Distribution is unlimited.

Copyright Notice

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

Abstract

The Activity Endpoint Profile is a profile on the EndPointReference returned by an OGSA Basic Execution Services CreateActivity call. It defines port-types that the endpoint must support, and values that MUST be returned from calls and values that MAY be returned. The goal is to provide a uniform mechanism to support requirements identified by the Production Grid Interoperability Working Group of the Open Grid Forum [cite]. Contents

[Abstract 1](#_Toc393200648)

[1 Introduction 4](#_Toc393200649)

[2 Notational Conventions 4](#_Toc393200650)

[3 Activity Endpoint Compliance Requirements 5](#_Toc393200651)

[3.1 RNS 1.1 Compliance 5](#_Toc393200652)

[3.2 EPR & Resource Property Metadata fields 5](#_Toc393200653)

[3.2.1 SupportsActivityEndpoint 5](#_Toc393200654)

[3.3 Activity Properties 5](#_Toc393200655)

[3.3.1 FactoryReference 5](#_Toc393200656)

[3.3.2 Stderr – Standard Error 6](#_Toc393200657)

[3.3.3 Stdout – Standard Output 6](#_Toc393200658)

[3.3.4 WorkingDirectoryReference 6](#_Toc393200660)

[3.3.5 ActivityDocument 6](#_Toc393200661)

[3.3.6 ActivityLogMachineReadable 6](#_Toc393200662)

[3.3.7 Status 6](#_Toc393200664)

[3.4 Required RNSEntry Lookup Elements 6](#_Toc393200666)

[3.4.1 Status 7](#_Toc393200667)

[3.4.2 ActivityLogHumanReadable 7](#_Toc393200668)

[3.4.3 ActivityLogMachineReadable 7](#_Toc393200669)

[3.4.4 ActivityDocument 7](#_Toc393200677)

[3.4.5 Stdout 7](#_Toc393200678)

[3.4.6 Stderr 7](#_Toc393200679)

[3.5 BESExtensionElement 7](#_Toc393200680)

[4 BES State Model Changes 7](#_Toc393200681)

[5 Optional Activity Endpoint Targets 10](#_Toc393200682)

[5.1 Activity Properties 10](#_Toc393200683)

[5.1.1 FileSystemEndpoint 10](#_Toc393200684)

[5.2 Optional RNSEntry elements returned from lookup 11](#_Toc393200685)

[5.2.1 WorkingDirectory 11](#_Toc393200686)

[5.2.2 AvailableFilesystems 11](#_Toc393200687)

[5.2.3 ActivityResourceUsage 11](#_Toc393200688)

[An RNSEntry that refers to a read-only ByteIO resource (file) that contains the execution resources consumed so far. The value returned is not guaranteed to be up-to-date. The document MUST conform to the OGF URF (Usage Record Format). 11](#_Toc393200689)

[5.2.4 ActivityProcessControl 11](#_Toc393200690)

[5.2.5 ProcMem 11](#_Toc393200691)

[5.2.6 Streams 12](#_Toc393200698)

[6 Extensions to the BES-Activity Port-type 12](#_Toc393200699)

[6.1 Operations 12](#_Toc393200705)

[6.1.1 GetActivityAttributesDocument 12](#_Toc393200706)

[6.1.2 GetActivityStatus 12](#_Toc393200707)

[6.1.3 ResumeActivity 13](#_Toc393200709)

[6.1.4 TerminateActivity 14](#_Toc393200710)

[6.1.5 GetActivityDocument 14](#_Toc393200712)

[6.1.6 ResumeActivity 15](#_Toc393200713)

[6.1.7 GetActivityHistory 15](#_Toc393200714)

[6.1.8 Purge 18](#_Toc393200715)

[7 Security Considerations 18](#_Toc393200739)

[8 Author Information 18](#_Toc393200740)

[9 Contributors 19](#_Toc393200741)

[10 Acknowledgements 19](#_Toc393200742)

[Full Copyright Notice 19](#_Toc393200743)

[Disclaimer 19](#_Toc393200744)

[Full Copyright Notice 19](#_Toc393200745)

[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 as references to the derived portions on all such copies and derivative works. The published OGF document from which such works are derived, however, 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 new or updated OGF documents in conformance with the procedures defined in the OGF Document Process, or as required to translate it into languages other than English. OGF, with the approval of its board, may remove this restriction for inclusion of OGF document content for the purpose of producing standards in cooperation with other international standards bodies. 20](#_Toc393200746)

[The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees. 20](#_Toc393200747)

[Intellectual Property Statement 20](#_Toc393200748)

[Normative References 20](#_Toc393200749)

# 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 AEP is a part of the second approach, profiling and extending existing specifications to meet the requirement. It combines, extends, and profiles six 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], OGSA-ByteIO WSRF Basic Profile 1.0 [GFD.98], and WS-Notification 1.3.

The OGSA Basic Execution Services specification (OGSA\_BES, or BES) [GFD.108] has been in use for over five years. Over the course of use several common extensions have been used by different implementers. In OGSA-BES the CreateActivity operation returns a WS-Addressing Endpoint Reference (EPR), which clients can subsequently use to refer to the new activity. While the specification requires that the EPR MUST be compliant with WS-Addressing EndpointReferenceTypes it makes no additional requirements.

The Activity Endpoint Profile is a profile on the EPR returned from CreateActivity. The profile

1. specifies that the OGSA-BES MUST return an EPR that implements the RNS 1.1 OGSA-WSRF Basic Profile 1.0 [GFD.172],
2. defines WS Addressing metadata fields that MUST be present in the EPR of the activity,
3. defines optional WS-Notification subscriptions,
4. defines a set of required RNS entries and optional RNS entries returned from the RNS lookup operation on compliant endpoints,
5. defines a set of resource properties exposing the activity properties, e.g., status,
6. and defines an activity port-type.

# 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 an “Activity Endpoint 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 2‑1: 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> |
| wsn | <http://docs.oasis-open.org/wsn/b-2> |
| aep | <http://schemas.ogf.org/aep/2012/03/aep> |
| aid | <http://schemas.ogf.org/jsdl/2010/06/activity-instance-description> |
| aid-ogf | <http://schemas.ogf.org/jsdl/2010/06/activity-instance-description-ogf> |

# Activity Endpoint Compliance Requirements

This section describes the compliance requirements.

## RNS 1.1 Compliance

Compliant implementations MUST implement RNS 1.1 and WSI-BSP and MAY support OGSA-WSRF Basic Profile 1.0.

## EPR & Resource Property Metadata fields

### SupportsActivityEndpoint

This **MANDATORY** Metadata entry in the Endpoint Reference for the activity endpoint and in the Resource Properties indicates whether the endpoint is compliant with the Activity Endpoint 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. The following pseudo schema illustrates this.

<wsa:EndpointReference>

...

<wsa:Metadata>

<aep:SupportsActivityEndpoint>

<xsd:boolean/>

</aep:SupportsActivityEndpoint>

</wsa:Metadata>

</wsa:EndpointReference>

## Activity Properties

This section refers to resource properties or attributes. This SHOULD be implemented using WS RF Resource Properties as defined in the OGSA-BP.

### FactoryReference

This attribute in the activity endpoint’s Resource Properties document is used to refer to the BES where the activity is executing. The attribute’s type is wsa:EndpointReferenceType, and it has a cardinality of exactly 1.

<aep:FactoryReference>

wsa:EndpointReferenceType

<aep:FactoryReference>

### Stderr – Standard Error

This property represents the name of the standard error file. The attribute’s type is xsd:string, and it has a cardinality of 0 to 1. This is the name of the file in working directory of the running job.

<aep:Stderr>xsd:string</aep:Stderr>

### Stdout – Standard Output

This property represents the name of the standard output file. The attribute’s type is xsd:string, and it has a cardinality of 0 to 1. This is the name of the file in working directory of the running job.

<aep:Stdout>xsd:string</aep:Stdout>

### WorkingDirectoryReference

This optional property represents a pointer to the activity’s working directory. The access to this location is implementation dependent. For instance it could be HTTP(S), GSIFTP, RNSEntry accessed through ByteIO. The attribute’s type is wsa:EndpointReferenceType and its cardinality is 1.

<aep:WorkingDirectoryReference>  
 wsa:EndpointReferenceType  
</aep:WorkingDirectoryReference > 0..\*

The cardinality of WorkingDirectoryReference is 0..\* to support multiple interfaces because different clients may understand different protocols.

### ActivityDocument

The ActivityDocument MUST be present. It refers to the BES activity document, for which semantics are already defined in the GFD.108.

### ActivityLogMachineReadable

This mandatory property represents the machine readable representation of the activity log. The format of this element is an instance of ActivityHistory element, which MUST be compliant with the Activity Instance Description [ref] specification. The ActivityHistory element encapsulates a sequence of ActivityHistoryEntry instances.

<aep:ActivityLogMachineReadable>

<aid:ActivityHistory>

<aid:ActiivtyHistoryEntry>..</aid:ActivityHistoryEntry>

<aid:ActiivtyHistoryEntry>..</aid:ActivityHistoryEntry>

</aid:ActivityHistory>

</aep:ActivityLogMachineReadable>

### Status

The ActivityStatus MUST be present. It refers to the BES activity status document, for which semantics are already defined in the GFD.108.

## Required RNSEntry Lookup Elements

The RNS 1.1 lookup operation MUST return the following RNSEntrys as defined in GFD.171 section 2.1.5 as a LookupResponse. Implementations MAY return additional elements as desired.

### Status

An RNSEntry that refers to a ByteIO resource (file), the content of which MUST be the ActivityStatus state model as defined in the OGSA-BES specification. Following the status this file MAY include implementation specific text such as a stack trace if the activity has failed.

### ActivityLogHumanReadable

An RNSEntry refers to a ByteIO resource (file).The content of which is the activity history in a human readable form. Note that the format of the file is implementation specific.

### ActivityLogMachineReadable

An RNSEntry refers to a ByteIO resource (file),the content of which is in XML format. The format of this element is defined in section 3.3.6

### ActivityDocument

An RNSEntry that refers to a ByteIO resource (file), the content of which is the ActivityDocument defined in the GFD .108.

*Stderr* –An RNSEntry that refers to a read-only StreamableByteIO resource (file). The contents of the file are the process’s output from “stderr”.

* *Stdout* – An RNSEntry that refers to a read-only StreamableByteIO resource (file). The contents of the file are the process’s output from “stdout”.

### Stdout

An RNSEntry that refers to a ByteIO resource (file), the content of which is the stdout of the activity as defined in 3.3.3. *Stdout* refers to a read-only StreamableByteIO resource (file). The contents of the file are the process’s output from “stdout”.

### Stderr

An RNSEntry that refers to a ByteIO resource (file), the content of which is the stderr of the activity as defined in 3.3.2. *Stderr*  refers to a read-only StreamableByteIO resource (file). The contents of the file are the process’s output from “stderr”.

## BESExtensionElement

Before submitting a job, a client application may want to be sure if the target BESFactory service is supporting Activity Endpoint Profile.In order to expose additional BESFactory capabilities GFD.108 already defines the bes:FactoryResourcesAttributesDocumentType/bes:BESExtension element. For instance this element may be useful for information services while it aggregates or discovers the capabilities of the registered BESFactory endpoints in a distributed computing infrastructure. This capability will be exposed through introducing the following mandatory BESExtension element.

<http://schemas.ggf.org/bes/2013/04/besExtensions/SupportActivityEndpointProfile>

# 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 [GFD.210]. In addition, there has been significant feedback from developers on the need to explicitly model pre-and-post-processing steps. This profile defines substates for the Pending and Running states. The profiled substates for Pending and Running are shown in Figures 2 and 3 below.

The “Held” substate of both Pending and Running is a special substate in that it corresponds to a set of substates, e.g., Pending:Created-Held, Pending:Meta-scheduling-Held, etc., where there is a \*-Held substate for each Pending and Running substate shown*except*Running:Queued. The substates of both Pending and Running are listed in order in Table 2. Not all substates are necessarily entered.

Held substates are only entered if the activity JSDL contains a “Hold” element as specified in Figure 2 of this document. Upon entering a Held state, processing of the activity by a compliant BES will stop until a ResumeActivites (Section 6.2.1) is received by the BES or the ResumeActivityporttype is called on the corresponding Activity Endpoint as defined in the AEP. Upon receipt of a resume, the compliant BES will move the activity to the next corresponding state, e.g., Pending:Staging-in from Pending:Scheduled-Held, and resume processing the activity.

Finally, the initial BES state model from GFD.108 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. Note that such a transition is consistent with the BES state model. An external observer may never see the Running state of an activity: the activity might appear to transition directly from Pending to Failed.

If you want to have a hold on an activity it will be indicated in the BES CreateActivity call.

<SHAHBAZ WILL PUT IN THE CREATEACTIVITY SCHEMA AND EXAMPLE.

The Hold sub-element of the JobProcessing element indicates when the BES should stop processing the activity and wait for a ResumeActivities as described in §3. The multiplicity is zero or more.

<HoldEnumeration>\*

HoldEnumeration

|  |  |
| --- | --- |
| Pending:Created | Running:Executing |
| Pending:Meta-scheduling | Running:Post-processing |
| Pending:Scheduled | Running:Staging-out |
| Pending:Staging-in |  |
| Pending:Pre-processing |  |

The semantics dictate that the activity move into the appropriate Held state AFTER exiting the specified state. Upon resumption, the activity will move into the subsequent state in the state model.

-------------- end of SHAHBAZ updates

Figure 2. AEP activity substates of Pending and Running. Note that there are many “Held” substates. Only one is being show for each. In Pending it illustrates a Held substate between Scheduled and Staging-in.

Table 2. Substates for Pending and Running in order that they are observed. Not all substates will be observed.

|  |  |
| --- | --- |
| Pending | Running |
| Created | Executing |
| Created-Held | Executing-Held |
| Meta-scheduling | Post-processing |
| Meta-scheduling-Held | Post-processing-Held |
| Scheduled | Staging-out |
| Scheduled-Held | Staging-out-Held |
| Staging-in |  |
| Staging-in-Held |  |
| Pre-processing |  |
| Pre-processing-Held |  |
| Queued |  |

# Optional Activity Endpoint Targets

## Activity Properties

### FileSystemEndpoint

If it is desired to provide a resource endpoint for activity file systems, as defined in the ActivityDocument, attributes addressing those endpoints SHOULD be included in the activity endpoint’s Activity Properties document. These attributes, if present, MUST include the following child elements.

For example, suppose that the activity’s JSDL specified a SCRATCH file system. Then the FileSystemEndpoint will describe the name of the file system, e.g., SCRATCH, the mode, and provide an EPR with which to interact with the file system.

#### FileSystemName

An element “FileSystemName” of type /xsd:string identifying the file system. The FileSystemName MUST be the same as that defined in the JSDL for the activity.

#### FileSystemMode

An element “FileSystemMode” of type /aep:FileSystemModeType indicating whether the specified file system is available for reading, writing, or both. /aep:FileSystemModeType is an enumeration of the /xsd:string values “read”, “write”, and “both”.

#### FileSystemEPR

An element “FileSystemEPR” of type /wsa:EndpointReferenceType addressing the endpoint is located.The cardinality of this element is 0 or more, but at least 1 of either FileSystemURI or FileSystemEPR MUST be present.

Example

<aep:FileSystemEndpoint>

<aep:FileSystemName> /xsd:string </aep:FileSystemName>

<aep:FileSystemMode> /aep:FileSystemModeType </aep:FileSystemMode>

<aep:FileSystemEPR> /wsa:EndpointReferenceType </aep:FileSystemEPR> \*

</aep:FileSystemEndpoint>

## Optional RNSEntry elements returned from lookup

OPTIONAL RNSEntry elements that MAY be present. If they are present though, they MUST have the following meaning and refer to the specified information.

### WorkingDirectory

An RNSEntry that refers to another RNS 1.1 endpoint listing the contents of the activity session directory where the activity is running. The *lookup()* operation on the WorkingDirectory will return a list of files and directories that are visible to the running job in the session directory. Reading, writing, creating, and deleting files and directories in the WorkingDirectory will read, write, create and destroy files and directories in the file system on which the job is running.

### AvailableFilesystems

At the option of the implementer there MAY be an RNSEntry that corresponds to each FileSystemEndpoint where the name of the RNSEntry is the FileSystemName element of the FileSystemEndpoint and the EPR of the RNSEntry is the FileSystemEPR of the FileSystemEndpoint. It is expected that the file system names are unique.

### ActivityResourceUsage

### An RNSEntry that refers to a read-only ByteIO resource (file) that contains the execution resources consumed so far. The value returned is not guaranteed to be up-to-date. The document MUST conform to the OGF URF (Usage Record Format).

### ActivityProcessControl

An RNSEntry that refers to a write-only ByteIO resource (file). Writing an integer to this file, in textual format, causes a Unix signal to be sent to the activity’s process. The signal sent will depend on the integer written: if the integer written is one of the possible values for a Unix signal, the signal sent will have that value. If the integer written is not a valid Unix signal, or a non-integer value is written, no signal will be sent to the process.

For example, if the string “9” is written to ActivityProcessControl, and signal 9 is supported by the underlying operating system, then signal 9 is sent to the process. If the string “SIGKILL” is written to the file, the input is ignored and no signal is issued.

### ProcMem

An RNSEntry that refers to a ByteIO resource (file) that provides access to the memory of sequential processes, so that debuggers can be attached. The ByteIO resource is an interface to the /proc/mem entry for the activity’s process, or equivalent interface for the local operating system. The contents of this file are operating system and implementation specific, as are the results of reading from and writing to the file, but they SHOULD be implemented to allow process debuggers to attach to the process in the same manner as if the process was running local to the debugger.

If the activity is a parallel job, such that multiple processes are involved, a single access point is typically insufficient for debugging the entire activity. In such cases, the semantics of this resource are undefined.

### Streams

An RNSEntry that refers to another RNS 1.1 endpoint that contains the following entries for input/output streams

* *Combined*-An RNSEntry that refers to a read-only StreamableByteIO resource (file). The contents of the file are the combined process’s output from “stdout” and “stderr”.
* *Stdin* – An RNSEntry that refers to a write-only StreamableByteIO resource (file). Data written to the file are streamed to the process as input from “stdin”.

*Exit Code*: The exit code of the activity when it terminates.

# Extensions to the BES-Activity Port-type

The BES specification provides an optional extension called the BES-Activity port-type for managing individual activities on a BES. This Profile mandates the implementation of that port-type on compliant systems, with an additional set of extensions.

## Operations

The BES specification did not include specific operations to implement with the port-type; we include a required set here.

### GetActivityAttributesDocument

This operation allows retrieving an instance of the activity’s resource properties element which includes all the elements of resource properties mentioned in Section 3.3.

#### Input(s)

This operation requires no input parameters.

#### Output(s)

* **GetActivityAttributesDocumentResponseType Response**: An XML document containing a /bes-aep:ActivityResourceAttributesDocument element, indicating the current state of the activity.

#### Fault(s)

* **UnknownActivityIdentifierFault**: The EPR for this activity is no longer valid (as when the activity has been removed from the BES).
* **NotAuthorizedFault**: The EPR for this activity is valid; however the validated user credential is not authorized to perform the specified operation. For instance, this activity may be owned by another user.
* **GenericFault:** The reason is implementation defined.

### GetActivityStatus

This operation allows a client to request the status of the activity.

#### Input(s)

This operation requires no input parameters.

#### Output(s)

* **GetActivityStatusResponseType Response**: An XML document containing a /bes:ActivityStatus element, indicating the current state of the activity.

#### Fault(s)

* **UnknownActivityIdentifierFault**: The EPR for this activity is no longer valid (as when the activity has been removed from the BES).
* **NotAuthorizedFault**: The EPR for this activity is valid; however the validated user credential is not authorized to perform the specified operation. For instance, this activity may be owned by another user.
* **GenericFault:** The reason is implementation defined.

### ResumeActivity

If the activity is in a Held state as defined in BES/JSDL Extensions 1.1, resume activity will transition the activity to the next appropriate state as defined in BES/JSDL Extensions 1.1 Table 2. For example, a ResumeActivity when the activity is in Staging-in-Held transitions the activity to Pre-Processing.

#### Input(s)

This operation requires no input parameters.

#### Output(s)

* **ResumeActivityResponseType Response**: An XML document containing a *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.

#### 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.
* **UnknownActivityIdentifierFault**: The EPR for this activity is no longer valid (as when the activity has been removed from the BES).
* **NotAuthorizedFault**: The EPR for this activity is valid; however the validated user credential is not authorized to perform the specified operation. For instance, this activity may be owned by another user.
* **CantApplyOperationToCurrentStateFault:** The activity is in a state from which the terminate operation cannot be applied. In the BES state model, this means that the activity is already in a terminal state (i.e. Finished, Failed, or Cancelled).
* **GenericFault:** The reason is implementation defined.

### TerminateActivity

This operation requests that the activity be terminated. Terminated corresponds to the canceled state in BES.. The BES attempts to terminate the activity. As a consequence of this operation, the activity MAY be terminated. If the activity cannot be terminated immediately, the eventual success of the operation must be determined through other operations (e.g. GetActivityStatus) or by subscribing to any generated events.

If the request is successful, the activity will eventually enter the *Cancelled* state. Invoking this operation on a Cancelled activity has no further effect.

#### Input(s)

This operation has no input parameters.

#### Output(s)

* **TerminateActivityResponse**: An XML document containing an element of type /bes:TerminateActivityResponse, indicating whether the BES successfully terminated the activity or not. The Terminated element is a boolean value indicating the BES successfully terminated (true) the activity or not (false). If false is returned then this is assumed that the activity cannot be terminated upon the incoming client request, thus it MAY eventually transition to the *Cancelled* state. If the activity no longer exists, or cannot be terminated then the operation must return a **CantApplyOperationToCurrentStateFault**.

<TerminateActivityResponse>  
<Terminated>xsd:Boolean</Terminated>?  
 |  
<SOAP-1.1:fault>…</SOAP-1.1:fault>?  
</TerminateActivityResponse>

#### Fault(s)

* **UnknownActivityIdentifierFault**: The EPR for this activity is no longer valid (as when the activity has been removed from the BES).
* **NotAuthorizedFault**: The EPR for this activity is valid; however the validated user credential is not authorized to perform the specified operation. For instance, this activity may be owned by another user.
* **CantApplyOperationToCurrentStateFault:** The activity is in a state from which the terminate operation cannot be applied. In the BES state model, this means that the activity is already in a terminal state (i.e. Finished, Failed, or Cancelled).
* **GenericFault:** The reason is implementation defined.

### GetActivityDocument

This operation requests the ActivityDocument description of the activity. As with the BES-Factory equivalent of this operation, this document may be different from that initially passed to the BES in the CreateActivity operation, as the BES may alter its contents to reflect policy or process within the service.

#### Input(s)

This operation requires no input parameters.

#### Output(s)

* **GetActivityDocumentResponseType Response**: The output from this operation is an ActivityDocumentResponse element containing the ActivityDocument for this activity.

#### Fault(s)

* **UnknownActivityIdentifierFault**: The EPR for this activity is no longer valid (as when the activity has been removed from the BES).
* **NotAuthorizedFault**: The EPR for this activity is valid; however the validated user credential is not authorized to perform the specified operation. For instance, this activity may be owned by another user.
* **GenericFault:** The reason is implementation defined.

### ResumeActivity

This operation requests that a HELD activity’s execution be resumed. In a BES which supports process suspension, calling ResumeActivity on an activity in the *Pendng:Held* or *Running:Held* state will transition the activity to the next state according to the state model*.* Invoking this operation on an activity in a non HELD state has no further effect.

#### Input(s)

This operation requires no input parameters.

#### Output(s)

* **ResumeActivityResponseType Response**: An XML document containing the response to the resume request. The Resumed element is a Boolean value indicating whether the BES successfully resumed the activity (true) or not (false). If true is returned, the activity is now in the appropriate state. If false is returned, the activity MAY eventually transition into the appropriate state. If the activity cannot be resumed then the activity MUST throw a fault.

<aep:ResumeActivityResponse>

<Resumed> xsd:boolean </Resumed> ?

|

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

<aep:ResumeActivityResponse>

#### Faults:

* **CantApplyOperationToCurrentStateFault:** The activity is in a state from which the Resume operation cannot be applied. In the extended BES state model above, this means that the activity is already in a terminal state (i.e. Finished, Failed, or Cancelled), or staging (i.e. Running:Stage-in or Running:Stage-out).
* **NotAuthorizedFault**: The EPR for this activity is valid; however the validated user credential is not authorized to perform the specified operation. For instance, this activity may be owned by another user.
* **OperationNotSupportedFault**:The EPR for this activity is valid; however, the BES on which this activity is running does not support activity Resume.

### GetActivityHistory

This operation requests the history events that have been issued for the activity.

#### Input(s)

* **GetActivityHistoryRequestType Request**: An XML document describing query parameters contained in the ActivityHistoryQuery element. If this element is not present in the GetActivityHistoryRequestType then BES SHOULD return all the activity’s log events happened until the time of request. Reasons for not returning the entire log might include the log file is too long.

<aep:GetActivityHistoryRequest>

<aep:ActivityHistoryQuery>bes-factory:ActivityStatusType</bes:ActivityStatus>+

<aep:ActivityStatusRangeQuery>

<StartState>|<EndState>|<ExactState>

</aep:ActivityStatusRangeQuery>

|

<aep:TimeRangeQuery>

<aep:Start>xsd:datetime</aep:Start> //date and time in milliseconds

<aep:End>xsd:datetime</aep:End> //date and time in milliseconds

</aep:GetActivityHistoryRequest>

To get all the activity logs in most of the cases is not scalable, therefore there should be an intuitive query mechanism to project activity logs in a simplified manner. In this section we defined a schema which provides an alternative query mechanism that enable client applications to specify multiple query parameters.

ActivityHistoryQuery: The parent element ActivityHistoryQuery contains all the complex query request encapsulating complex query data types. The multiplicity of this element is one.

<ActivityHistoryQuery>

<KindOfQuery>..</KindOfQuery> +

<WithIterator>Boolean</WithIterator> +

</ActivityHistoryQuery>

KindOfQuery: This element defines a data type of different kind of queries which includes time, activity status, negation, and recursive join of these kind of queries.

<KindOfQuery>

<TimeQuery> |

<StatusQuery> |

<JoinQuery> |

<NegationQuery> |

</KindOfQuery>

TimeQuery: The TimeQuery element specifies start and end time of activity.

<TimeQuery>

<Begin>xsd:dateTime</Begin>

<End>xsd:dateTime</End>

</TimeQuery>

For this element the Begin value must be less than the End one.

StatusQuery: The StatusQuery represents the higher and lower bound of activity status. The activity status MUST be compliant with the OGSA-BES, HPC-BP, and HPC-FSP profile.

<StatusQuery>

<StartState>bes:ActivityStatus</StartState>0..1

<EndState>bes:ActivityStatus</EndState>0..1

</StatusQuery>

This profile mandates that the StatusQuery element, if specified in a query request, MUST contain either StartState or EndState.

NegationQuery: This element defines negation of KindOfQuery, which in terms of logic is expressed as NOT or complement operator.

<NegationQuery>

<Operand>aep:KindOfQuery</Operand>1

</NegationQuery>

JoinQuery: This complex data type allows to construct the pair of KindOfQuery elements with AND and OR Boolean operators. By using this construct rich queries to specify complex KindOfQuery instances.

<JoinQuery>

<LeftOperand>aep:KindOfQuery</LeftOperand> 1

<RightOperand>aep:KindOfQuery</RightOperand> 1

<Operator>AND | OR</Operator> 1

</JoinQuery>

The use of LeftOperand, RightOperand and Operator elements MUST be present if JoinQuery is intended to be used.

In order to proof our schema, here we present a scenario as an example where the query is return activity’s history events between Running:Executing and Finished.

Example 1:

<GetActivityHistoryRequest>

<StatusQuery>

<StartState>

<bes:ActivityState>Running:Executing</bes:ActivityState>

</StartState>

<EndState>

<bes:ActivityState>Finished</bes:ActivityState>

</EndState>

</StatusQuery>

</GetActivityHistoryRequest>

Example 2:

Assume a more complex example in which set of events to be fetched when the job is not Finished AND the time duration is between 01:00 PM to 02:00PM.

<GetActivityHistoryRequest>

<JoinQuery JoinOperator=”OR”>

<LeftOperand xsi:KindOfQueryType>

<TimeQuery>

<StartTime>13:00:20</StartTime> //TODO

<EndTime>14:00:20</EndTime> //TODO

</TimeQuery>

</LeftOperand>

<RightOperand xsi:type=”KindOfQueryType”>

<Negation>

<operand xsi:type=”KindOfQueryType”>

<StatusQuery>

<bes:ActivityState>Finished</bes:ActivityState>

</StatusQuery>

</operand>

</Negation>

</RightOperand>

</JoinQuery>

</GetActivityHistoryRequest>

#### Output(s)

* **GetActivityHistoryResponseType Response**: In response this method returns an ActivityHistory element containing an array of ActivityHistoryEntry elements. If the activity cannot be located, or the corresponding activity’s history cannot be returned then the GetActivityHistoryResponse MUST contain a SOAP-1.1 fault element instead of ActivityHistory elements.

<aep:GetActivityHistoryResponse>

<aid:ActivityHistory>

<aid:ActivityHistoryEntry timestamp=”xsd:datetime”>

<bes:ActivityStatus>..<bes:ActivityStatus>

<aid:Events>?

<aid:Event timestamp=”xsd:datetime”>..</aid:Event>+

</aid:Events>

</aid:ActivityHistoryEntry>

<aid:ActivityHistory>

<aep:GetActivityHistoryResponse>

While invoking this method if client requests for history events associated with anon-existent activity statusthen the server MUST reply with soap 1.1 fault element stating reason as UnknownActivityStatusFault respective to the activity status information being returned.

#### Fault(s)

* **UnknownQueryFault:** The query could not be interpreted, or failed to execute for some reason.
* **NotAuthorizedFault**: The EPR for this activity is valid; however the validated user credential is not authorized to perform the specified operation. For instance, this activity may be owned by another user.

# Security Considerations

Access control is out of scope.

It is presumed that all communication is conveyed over a secure channel such as TLS or https. Implementations are encouraged to monitor for security fault patterns.

# Author Information

Daniel Dougherty

University of Virginia

Andrew Grimshaw (editor)  
University of Virginia

Shahbaz Memon  
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

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.

# Disclaimer

This document and the information contained herein is provided on an “As Is” basis and the OGF 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.

# Full Copyright Notice

Copyright (C) Open Grid Forum (2014). Some 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 as references to the derived portions on all such copies and derivative works. The published OGF document from which such works are derived, however, 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 new or updated OGF documents in conformance with the procedures defined in the OGF Document Process, or as required to translate it into languages other than English. OGF, with the approval of its board, may remove this restriction for inclusion of OGF document content for the purpose of producing standards in cooperation with other international standards bodies.

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

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]

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

WS-Notification