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: Activity Credential. Distribution is unlimited.

Copyright Notice

Copyright © Open Grid Forum (2012). 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](#_Toc326939025)

[1 Introduction 4](#_Toc326939026)

[2 Notational Conventions 4](#_Toc326939027)

[3 Activity Endpoint Compliance Requirements 5](#_Toc326939028)

[3.1 RNS 1.1 Compliance 5](#_Toc326939029)

[3.2 EPR Metadata fields 5](#_Toc326939030)

[3.2.1 SupportsActivityEndpoint 5](#_Toc326939031)

[3.3 Resource Properties 5](#_Toc326939032)

[3.3.1 SupportsActivityEndpoint 5](#_Toc326939033)

[3.3.2 BESFactoryEPR 5](#_Toc326939034)

[3.4 Required RNSEntry Elements returned from lookup 6](#_Toc326939035)

[3.4.1 Status 6](#_Toc326939036)

[3.4.2 HistoryHuman 6](#_Toc326939037)

[3.4.3 HistoryMachine 6](#_Toc326939038)

[3.4.4 ActivityDocument 6](#_Toc326939039)

[3.5 JSDL Resource Element 6](#_Toc326939040)

[4 Optional Activity Endpoint Compliance Targets 7](#_Toc326939041)

[4.1 Resource Properties 7](#_Toc326939042)

[4.1.1 FileSystemEndpoint 7](#_Toc326939043)

[4.1.2 SSHEndpoint 7](#_Toc326939044)

[4.1.3 GSI-SSHEndpoint 7](#_Toc326939045)

[4.2 Optional RNSEntry elements returned from lookup 8](#_Toc326939046)

[4.2.1 SessionDirectory 8](#_Toc326939047)

[4.2.2 OGSAResources 8](#_Toc326939048)

[4.2.3 ControlFile 8](#_Toc326939049)

[4.2.4 ProcMem 8](#_Toc326939050)

[4.2.5 GridFTPEndpoint 8](#_Toc326939051)

[4.2.6 HTTPEndpoint 8](#_Toc326939052)

[4.2.7 Streams 9](#_Toc326939053)

[5 Extensions to the BES-Activity Port-type 9](#_Toc326939054)

[5.1 Extended BES State Model 9](#_Toc326939055)

[5.2 Attributes 9](#_Toc326939056)

[5.3 Operations 9](#_Toc326939057)

[5.3.1 GetActivityStatus 10](#_Toc326939058)

[5.3.2 TerminateActivity 10](#_Toc326939059)

[5.3.3 GetActivityDocument 10](#_Toc326939060)

[5.3.4 SuspendActivity 11](#_Toc326939061)

[5.3.5 ResumeActivity 12](#_Toc326939062)

[5.3.6 GetActivityHistory 12](#_Toc326939063)

[5.3.7 Destroy 13](#_Toc326939064)

[6 WS-Notification Subscriptions - Optional 13](#_Toc326939065)

[7 Security Considerations 14](#_Toc326939066)

[8 Author Information 14](#_Toc326939067)

[9 Contributors 15](#_Toc326939068)

[10 Acknowledgements 15](#_Toc326939069)

[Full Copyright Notice 15](#_Toc326939070)

[Intellectual Property Statement 15](#_Toc326939071)

[Normative References 16](#_Toc326939072)

# Introduction

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 returns 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 capability, e.g., status,
6. and defines an activity management 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”.

Matrix of used specs and their maturity.

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> |
| wsn | <http://docs.oasis-open.org/wsn/b-2> |
| aep | <http://schemas.ogf.org/aep/2012/03/aep> |

# Activity Endpoint Compliance Requirements

This section describes the compliance requirements.

## RNS 1.1 Compliance

Compliant implementations MUST also implement RNS 1.1, and OGSA-WSRF Basic Profile 1.0.

## EPR Metadata fields

### SupportsActivityEndpoint

This Metadata entry in the Endpoint Reference for the activity endpoint 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.

<wsa:EndpointReference>

 ...

 <wsa:Metadata>

 <aep:SupportsActivityEndpoint>

 /xsd:boolean

 </aep:SupportsActivityEndpoint>

 </wsa:Metadata>

</wsa:EndpointReference>

## Resource Properties

Should these be a Resource Property and/or a BES-Factory Attribute?

### SupportsActivityEndpoint

This attribute in the activity endpoint’s Resource Properties document indicates whether the endpoint is compliant with the Activity Endpoint Profile. The attribute’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 attribute indicates non-compliance.

<aep:SupportsActivityEndpoint>

 /xsd:boolean

</aep:SupportsActivityEndpoint>

### BESFactoryEPR

This attribute in the activity endpoint’s Resource Properties document can be 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:BESFactoryEPR>

 /wsa:EndpointReferenceType

<aep:BESFactoryEPR>

## Required RNSEntry Elements returned from lookup

### Status

An RNSEntry that refers to a ByteIO resource (file), the content of which is the ActivityStatus as defined in OGSA-BES and implementation specific text such as a stack trace if the activity has failed.

### HistoryHuman

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

### HistoryMachine

An RNSEntry that refers to a ByteIO resource (file), the content of which is in XML format.

Format is TBD, Mike, Bastian, Shabaz to be the "deciders" <Activity Instance Document spec looks promising, need to see where it is in the approval process.>

We recommend using an XML rendering of the History Events from section 6 for the machine history. Essentially, this will result in a way to retrieve the same job history either streamed as the job runs (through notifications), or requested all in one piece (through RNS or GetHistory function).

### ActivityDocument

An RNSEntry that refers to a ByteIO resource (file), the content of which is the ActivityDocument defined in the BES specification. <Need to agree on the schema of the activity document.>

## JSDL Resource Element

When a user creates a Job, they may wish to specify AEP compliance as a matching parameter for the target BES. This will be accomplished through the use of a new JSDL element,

**/jsdl:JobDefinition/jsdl:JobDescription/jsdl:Resources/aep:SupportsActivityEndpoint**

The element’s type is /xsd:boolean. An element value of “true” indicates that the job requires an AEP compliant BES, a value of “false” or the absence of such element indicate that the job has no requirement regarding AEP compliance.

<jsdl:JobDefinition>

 <jsdl:JobDescription>

 …

 <jsdl:Resources>

 …

 <aep:SupportsActivityEndpoint>

 /xsd:boolean

 </aep:SupportsActivityEndpoint>

 </jsdl:Resources>

 </jsdl:JobDescription>

</jsdl:JobDefinition>

# Optional Activity Endpoint Compliance Targets

## Resource 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 Resource Properties document. These attributes, if present, MUST include the following child elements.

#### 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”.

#### FileSystemURI

An element “FileSystemURI” of type /xsd:anyURI indicating where 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. Note that the list of URI’s for session directories is not necessarily fixed over the lifetime of the activity.

#### 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. Note that the list of EPR’s for session directories is not necessarily fixed over the lifetime of the activity.

<aep:FileSystemEndpoint>

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

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

 <aep:FileSystemURI> /xsd:anyURI </aep:FileSystemURI> \*

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

</aep:FileSystemEndpoint>

### SSHEndpoint

What are we using SSH to connect to?

<aep:SSHEndpoint>

 /wsa:EndpointReferenceType

</aep:SSHEndpoint>

### GSI-SSHEndpoint

What are we using GSI-SSH to connect to?

<aep:GSI-SSHEndpoint>

 /wsa:EndpointReferenceType

</aep:GSI-SSHEndpoint>

## Optional RNSEntry elements returned from lookup

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

### SessionDirectory

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 SessionDirectory 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 SessionDirectory will read, write, create and destroy files and directories in the file system on which the job is running.

### OGSAResources

An RNSEntry that refers to a read-only ByteIO resource (file) that contains the execution resources consumed, not guaranteed to be up-to-date - should this be modeled Usage Record?

### ControlFile

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 ControlFile, 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.

### GridFTPEndpoint

Should these be Resource Properties instead of RNS entries?

An RNSEntry that refers to a ByteIO resource (file) that contains an EPR addressing the GridFTP endpoint for the activity. The EPR is a degenerate and contains only the URI of the GripFTP endpoint.

### HTTPEndpoint

Should these be Resource Properties instead of RNS entries?

An RNSEntry that refers to a ByteIO resource (file) that contains an EPR addressing the HTTP endpoint for accessing the activity’s session directory. The EPR is a degenerate and contains only the URI of the HTTP session directory endpoint.

### Streams

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

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

# 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.

## Extended BES State Model

The following state transition diagram is an extension of the basic BES state diagram. It supports state transitions for file staging and Suspend/Resume on the activity. Note that this extension follows the requirements in the BES specification for conformant extensions.


## Attributes

All of the attributes specified for the BES-Activity port-type in the BES Specification MUST be included.

## Operations

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

### 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.

### TerminateActivity

This operation requests that the activity be terminated. 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)

* **TerminateActivityResponseType Response**: An XML document containing an element of type /bes:TerminateActivityResponse, indicating whether the BES successfully terminated the activity or not. If the activity no longer exists, or cannot be terminated then the TerminateActivityResponse element MUST contain a SOAP-1.1 fault element instead of a Terminated element.

#### 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).

### 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.

### SuspendActivity

This operation requests that the activity’s execution be suspended. In a BES which supports process suspension, calling SuspendActivity on an activity in the *Running:Executing* state will transition the activity to the *Running:Suspended* state. Invoking this operation on a *Running:Suspended* activity has no further effect.

#### Input(s)

This operation requires no input parameters.

#### Output(s)

* **SuspendActivityResponseType Response**: An XML document containing the response to the suspend request. The Suspended element is a Boolean value indicating whether the BES successfully suspended the activity (true) or not (false). If true is returned, the activity is now in the *Running:Suspended* state. If false is returned, the activity MAY eventually transition into the *Running:Suspended* state. If the activity no longer exists, or cannot be suspended then the SuspendActivityResponse element MUST contain a SOAP-1.1 fault element instead of a Suspended element.

<aep:SuspendActivityResponse>

 <Suspended> xsd:boolean </Suspended> ?

 |

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

<aep:SuspendActivityResponse>

#### Faults:

* **CantApplyOperationToCurrentStateFault:** The activity is in a state from which the suspend 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).
* **SuspendNotSupportedFault:** The EPR for this activity is valid; however, the BES on which this activity is running does not support activity Suspend/Resume.
* **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.

### ResumeActivity

This operation requests that a suspended activity’s execution be resumed. In a BES which supports process suspension, calling ResumeActivity on an activity in the *Running:Suspended* state will transition the activity to the *Running:Executing* state. Invoking this operation on a *Running:Executing* activity 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 *Running:Executing* state. If false is returned, the activity MAY eventually transition into the *Running:Executing* state. If the activity no longer exists, or cannot be resumed then the ResumeActivityResponse element MUST contain a SOAP-1.1 fault element instead of a Resumed element.

<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).
* **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.

### GetActivityHistory

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

#### Input(s)

* **GetActivityHistoryRequestType Request**: An XML document containing the requested detail level of events that the BES should return for the activity. If no EventLevel element is included, a default value <what should be the default> will be applied.

<aep:GetActivityHistoryRequest>

 <aep:EventLevel>

 /xsd:integer

 </aep:EventLevel> ?

</aep:GetActivityHistoryRequest>

#### Output(s)

* **GetActivityHistoryResponseType[] Response**: A vector of ActivityHistoryEvent elements for the activity. Each ActivityHistory element contains the EventLevel of the entry, as well as the contents of the event <To be defined>. If the activity cannot be located, or the history events cannot be returned then the GetActivityHistoryResponse MUST contain a SOAP-1.1 fault element instead of ActivityHistory elements.

<aep:GetActivityHistoryResponse>

 <aep:ActivityHistoryEvent>

 <aep:EventLevel> /xsd:integer </aep:EventLevel>

 <aep:HistoryEvent> … </aep:HistoryEvent>

 <aep:ActivityHistoryEvent> \*

 |

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

<aep:GetActivityHistoryResponse>

#### 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.

### Destroy

\*\*WSRF-BP, section 5.1 mandates the use of the destroy mechanism for compliant services, hence this function would be redundant.\*\*

Analogous to “purge” on the factory.

Removes all state information associated with the activity, working directories, status, etc. Subsequent invocations on the EPR of the activity will fail, and the activity will no longer appear in lists of activities on the containing OGSA-BES resource.

# WS-Notification Subscriptions - Optional

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

How? 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.

**Topic: State Change**

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

Topic:

/aep:ActivityStateChanged

/aep:ActivityStateChanged/ReachedFinalState

Returns:

<aep:BESActivityStateChangedContents>

 /wsa:EndpointReferenceType

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

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

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

</aep:BESActivityStateChangedContents>

**Topic: History event with a level of detail**

These will warrant further discussion. What do the topics mean? Are they hierarchical, such that subscribing to topic Level2 means you receive notification from Level1 and Level0 as well? Or should we require a subscription for each level the client wishes to receive? Should we specify exactly what events are possible, and at which level each event occurs?

Multiple topics, one for each history level. Level 0 is most coarse. Another way to think of it is as different events. Notification contains the history event.

Topic:

/aep:ActivityHistoryEvent

/aep:ActivityHistoryEvent/EventLevel0

/aep:ActivityHistoryEvent/EventLevel1

/aep:ActivityHistoryEvent/EventLevel2

/aep:ActivityHistoryEvent/EventLevel3

/aep:ActivityHistoryEvent/EventLevel4

Returns:

<aep:BESActivityHistoryEventContents>

 <aep:EndpointReference>

 /wsa:EndpointReferenceType

 <aep:EndpointReference>

 <aep:HistoryEvent>

 /xsd:string

 </aep:HistoryEvent>

 <aep:EventLevel>

 /xsd:integer

 </aep:EventLevel>

</aep:BESActivityHistoryEventContents>

# Security Considerations

Access control is out of scope.

# Author Information

Bastian Demuth
Forschungszentrum Juelich (FZJ)

Daniel Dougherty

University of Virginia

Andrew Grimshaw (editor)
University of Virginia

Michael Saravo
University of Virginia

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 the readability and accuracy of this document.

Full Copyright Notice

Copyright © Open Grid Forum (2012). 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 GGF 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]

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

WS-Notification