

Template for Interface Specification

Authors	Loukik Kudarimoti
Date	4-12-06
Current Version	1.3

Document Change Log

As SA3-WI15 Document			
Version number	Date	Description of change	People
1.0	30-11-06	First draft issued	Loukik Kudarimoti
1.1	30-11-06	Some comments added	Szymon Trocha
1.2	30-11-06	Addressed comments, made some changes based on comments	Loukik Kudarimoti
1.3	4-12-06	Added Naming convention for service name	Loukik Kudarimoti

Table of Contents

1. INTRODUCTION.....	4
2. NAMING CONVENTION TO BE FOLLOWED FOR YOUR DOCUMENTS.....	4
3. EXPLANATION OF TEMPLATE FOR INTERFACE SPECIFICATION DOCUMENT	4
3.1. GENERAL INFORMATION	5
3.2. FUNCTIONALITY.....	5
3.3. APPENDICES	7
4. TEMPLATE FOR INTERFACE SPECIFICATION DOCUMENT	7
1. GENERAL INFORMATION	7
2. FUNCTIONALITY - <PROVIDE FUNCTIONALITY NAME HERE>.....	7
APPENDIX I.....	10

1. Introduction

The intention behind an Interface Specification document is to accurately define the various interfaces available on a perfSONAR service. Such a definition includes detailed information about all the request messages, response messages, notifications that are supported by the service.

An interface specification document is very vital to many entities within the project such as the testing teams who will be independently testing the different perfSONAR services, the developers of analysis tools who will need accurate description of interfaces and also for other developers involved in perfSONAR development. It is then vital for such a document to be maintained and once the document is made public, any changes be appropriately logged and all users of the document be notified.

Interface specifications are usually grouped based on the functionality within the perfSONAR service that they try to provide via the messages. For example, if a particular perfSONAR service can cook some curry and also bake a pizza, messages will have to be defined for each of the functionalities i.e. for cooking curry, this is how you should request and these are the parameters (curry name, spice level, etc) that you should provide. Similarly, for baking a pizza, appropriate messages which will result in the service starting to bake a pizza of choice will need to be provided. (Of course, perfSONAR services are not yet capable of doing such tasks). Hence, it makes sense to establish a relationship between an Interface specification document and a Functionality specification document. All interface descriptions should thus be grouped under the functionality that they provide.

2. Naming convention to be followed in your documents

<code>service-type</code>	= Possible values are: MA, MP, LS, AA, ToS
<code>service-version</code>	= A number
<code>namespace-prefix</code>	= Namespace prefix associated with namespace name
<code>namespace-URI</code>	= URI reference identifying the namespace name

3. Explanation of Template for Interface specification document

This section describes how to fill the Interface specification file template. It specifies how information should be grouped, which information should be provided and in what format.

Please make sure that all the documents that you write have a 'Title Page' and 'Document Change Log' as you have seen in this document.

3.1. General Information

Service Name

You MUST specify your service name here.

You MUST use the following Naming convention:

```
<unique_service_name>_Type_<service_type>-  
                                <service_version>
```

Example: RRD-Type_MA_1.1

Service Type

You MUST specify type of your service here.

Version/release

You MUST specify service version number or release number.

Service Description

You MUST write some description of your service here.

Contact person (s)

You MAY specify a contact person(s) for this service so that other users can contact in case of questions.

Contact Information

You MUST specify some contact (preferably e-mail address) either with the above contact person or with your organisation.

3.2. Functionality

The Functionality sections list the functionalities supported by the service and provide detailed information about the interfaces with which the functionalities can be made use of.

There should be **at least one** 'Functionality' section in the document. There **MUST** be as many numbered 'Functionality' sections in the document as there are features available on the service software.

Each Functionality section's contents should start off with some short introduction of on the functionality itself. The objective of this introductory phrase is to provide a high level overview of what the software actually does and a high level overview of the request to be given and response to be expected.

For each 'Functionality' section in the document, there should be at least one Request Message and one Response Message. Each Request message should have a unique name/id. Each Response message should reference to at least one existing unique name/id of a Request message.

For example: If the request message is

```
Request Message - make_pizza_margharita_1
```

One of the Response messages for it could be

```
Response Message - your_pizza_margharita_1 for:  
make_pizza_margharita_1
```

Each Request and Response message should provide a pointer to a .rnc (Relax NG Compact) file. All .rnc files should be added to the Appendix. Relax NG compact is quite simple. A tutorial can be found at: <http://relaxng.org/compact-tutorial-20030326.html>

All Request and Response messages **MUST** contain an explanation of the .rnc file attached in the appendix. Such an explanation should contain information about the semantics of the data expected in these messages as well as any explanations about the structure/logic etc. that might aid the reader in understanding the request messages.

Notification messages can also be specified with the help of the template. The specification should contain information about the direction of the notification – whether it is outbound from the service or inbound into the service. The .rnc file should be present in the appendix and its explanation should be provided.

Examples for each type of Request and Response messages **MUST** be provided in the Examples section. A reference containing the unique name of the message should also be provided. Examples **MUST** be in XML. The XML should conform to the .rnc schema specification. Examples should be commented where possible (using xml structure for comments)

The Other Information section **MAY** be used to provide any other additional information that might be of help to the reader. Such information could be for example: The service will do a special action if a message without data block is present.

3.3. *Appendices*

The Appendices contains as many sections as there are Interfaces mentioned in the document i.e. for each Interface listed in the document, a corresponding Appendix should be present. Each Appendix section **MUST** contain .rnc files for each message specified for the Interface. The header for each .rnc file should contain the unique name assigned for that request or response message.

4. Template for Interface Specification Document

This section presents the template for the interface specification document. You **MUST** use it while writing a guide to your own interface specification file.

Fields in brackets "<"/>" are placeholders for text written by developer. You **MUST** delete them while editing the template. For naming convention used see section 2.

NOTE: Please make sure that all the documents that you write have a 'Title Page' and 'Document Change Log' as you have seen in this document.

--- Template beings here ---

1. General Information

Service Name:	<code><service-name></code>
Service Type:	<code><service-type></code>
Version/release:	<code><service-version></code>
Service Description:	<code><service-description></code>
Contact person(s):	<code><contact-person></code>
Contact Information:	<code><contact-information></code>

2. Functionality - `<provide functionality name here>`

Introduction

`<Introduction to functionality>`

Number of request messages supported: `<no. of messages>`

Number of possible error-free response message types: <no. of messages>

Request Message - <unique-request-name/message-id>

<Provide explanation of different sections of the .rnc file>

<Provide pointer to appropriate appendix containing .rnc
(Relax-NG Compressed file)>

**Response Message - <unique-response-message-name/id> for:
<reference-to-request-name/message-id>**

<Provide explanation of different sections of the .rnc file>

<Provide pointer to appropriate appendix containing .rnc
(Relax-NG Compressed file)>

Request Message - <unique-request-name/message-id>

<Provide explanation of different sections of the .rnc file>

<Provide pointer to appropriate appendix containing .rnc
(Relax-NG Compressed file)>

**Response Message - <unique-response-message-name/id> for:
<reference-to-request-name/message-id>**

<Provide explanation of different sections of the .rnc file>

<Provide pointer to appropriate appendix containing .rnc
(Relax-NG Compressed file)>

Notification Message - <unique-notification-message-name/id>

Notification Direction – <Service Sends this message/
Service receives this message>

<Provide explanation of different sections of the .rnc file>

<Provide pointer to appropriate appendix containing .rnc
(Relax-NG Compressed file)>

Examples

<Example-Name>
 <reference to unique message name/message-id>
 <Short explanation of example>
 <Example-xml>

<Example-Name>
 <reference to unique message name/message-id>
 <Short explanation of example>
 <Example-xml>

Other Information

Appendix I

<unique-request-message-name/message-id present in Functionality (1)>
< .rnc file for this unique-request-message-name/message-id >

<unique-response-message-name/message-id present in Functionality (1)>
< .rnc file for this unique-response-message-name/message-id >

--- Template ends here ---