Template for Metadata Configuration File

Authors	Loukik Kudarimoti, Szymon Trocha
Date	04-12-2006
Current Version	1.4

Document Change Log

As SA3-WI15 Document			
Version number	**Date**	**Description of change**	**People**
1	10-11-06	First draft issued	Szymon Trocha
1.1	10-11-06	Provided some content to ‘Other information’ section.	Loukik Kudarimoti
1.2	23-11-06	Added elements level numbering.	Szymon Trocha
1.3	01-12-06	Changed structure to reflect actual metadata config file description document	Szymon Trocha
1.4	4-12-06	Added naming convention to be followed for service name	Loukik Kudarimoti

Table of Contents

1. Introduction

XML metadata configuration file describes measurement data stored in Measurement Archives. It consists of blocks: Metadata (one or more) and Data. Individual sections must have unique identifiers (meta1/data1,..., metaN/dataN).

This document provides a specification for preparation of description of metadata configuration file. It describes obligatory parts and provides a template for a description of metadata configuration file, which must be filled by a service developer. Such description will detail all fields used within the file and will help the developers and users to properly configure and run the service.

2. Naming convention to be followed in your documents

`service-type`	= Possible values are: MA, MP, LS, AA, ToS
`service-version`	= A version number
`namespace-prefix`	= Namespace prefix associated with namespace name
`namespace-URI`	= URI reference identifying the namespace name
`n.n.n.n.n`	= Level number in XML block as given in the example

3. The *metadata.configuration* file

3.1. Explanation of Template for sample metadata.configuration file

This section describes how to fill the metadata configuration file template. It specifies which information should be provided and in what format.

3.1.1. 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.

3.1.2. Introduction

You MUST explain here the role and usage of this configuration file. Please try to explain the rationale behind the organisation of the metadata blocks and also provide an explanation of the key.

You MUST describe here the structure of this document so that the user can navigate through it easily.

3.1.3. Overview of the configuration file

You MUST give an overview of different blocks within the configuration file. Make sure the overview goes with the given example(s) below. The purpose is to give the user a general understanding of the structure of the configuration file. Also describe the way a user has to modify the file to reflect his configuration. Its important to specify which sections can be repeated as many as necessary. While describing the configuration file refer to element numbers used in the example below which indicate levels they are nested within the file.

3.1.4. Examples

This section MUST contain one or more examples for each of the configuration file. It doesn’t have to contain all blocks of the file but its subset large enough to explain and understand file structure. This will help in better understanding of the configuration file. While giving an example use levels numbering which uniquely identify all elements within the example file. See example of such numbering below:

```
1 <nmwg:store
        xmlns:nmwgt="http://ggf.org/ns/nmwg/topology/2.0/"
        xmlns:nmtm="http://ggf.org/ns/nmwg/time/2.0/"
        xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
        xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
        xmlns:netutil="http://ggf.org/ns/nmwg/characteristic/utilization/2.0/"
        xmlns="http://ggf.org/ns/nmwg/base/2.0/">

1.1     <nmwg:metadata id="meta1">
1.1.1     <netutil:subject id="subj1">
1.1.1.1     <nmwgt:interface>
1.1.1.1.1     <nmwgt:hostName>mordor.middleearth.pl</nmwgt:hostName>
1.1.1.1.2     <nmwgt:ifAddress type="ipv4">10.1.2.3</nmwgt:ifAddress>
1.1.1.1     </nmwgt:interface>
1.1.1     </netutil:subject>
1.1.2     <nmwg:parameters>
1.1.2.1     <nmwg:parameter
                 name="supportedEventType">utilization</nmwg:parameter>
1.1.2     </nmwg:parameters>
1.1     </nmwg:metadata>

1.2     <nmwg:data id="data1" metadataIdRef="meta1">
1.2.1     <nmwg:key>
1.2.1.1     <nmwg:parameters>
1.2.1.1.1     <nmwg:parameter
                 name="file">/ps/data/rrd/test/test.rrd</nmwg:parameter>
1.2.1.1.2     <nmwg:parameter name="dataSource">bytes</nmwg:parameter>
1.2.1.1.3     <nmwg:parameter name="valueUnits">Bps</nmwg:parameter>
1.2.1.1     </nmwg:parameters>
1.2.1     </nmwg:key>
1.2     </nmwg:data>
```

3.1.5. Namespaces

Formatted: Bullets and Numbering

XML namespaces provide a method for qualifying element and attribute names used in XML documents by associating them with namespaces identified by URI references.

The metadata configuration file defines namespaces in `store` element. All used namespaces MUST be defined in metadata configuration file description.

The following schema should be used for describing namespaces:

`<namespace-prefix>=<namespace-URI>`

3.1.6. Schema elements explained

Formatted: Bullets and Numbering

This part of metadata configuration file contains a list of all elements used within the file together with numbers indicating their levels within the XML block. Numbers correspond to example given above.

In general the metadata configuration file consists of some elements which are just building blocks of the structure of this file while the other contain useful users data. You MUST specify all elements but it’s enough to write that this element is a container element. The schema elements explanation part consists of two blocks repeated as many times as it is necessary to list all elements:

Element-name$^+$
  Attribute*

($^+$) means that this block is mandatory for each element and appears at least once in the explanation
(*) means that this block is optional and is described only when the element has some attributes

The following schema should be used for describing elements of metadata configuration file:

`n.n.n.n.n` **Element-name:** `<namespace-prefix>:<meta-subject-element-name>`

Mandatory

You MUST specify here if this element is mandatory in the metadata configuration file (YES/NO).

Description

Here you MUST describe the element meaning.

Possible values*

Here you MAY specify possible values if it comes from a set a few restricted values.

Default value*

You MAY specify here default values for this element if any.

Constraints (formatting, etc) on values*

Here you MAY describe constraints for the element value.

Error Information if property is missing/wrong value*

Here you MAY specify error information generated when this element is missing or wrong. This helps the user in proper debugging and it helps us in verifying the completeness and robustness of the software code.

Fields marked with (*) are optional. If the element is a container element or specifying more fields isn’t necessary one can omit them.

If any of the elements of metadata configuration file uses `attribute` element the following schema should be used for describing it:

Attribute: `<attribute-name>`

Mandatory

You MUST specify here if this attribute is mandatory in the metadata configuration file (YES/NO).

Description

Here you MUST describe attribute meaning.

Values

Here you MUST specify possible values if it comes from a set a few restricted values or describe how the values are created.

Default value*

You MAY specify here default values for this attribute if any.

Constraints (formatting, etc) on values*

Here you MAY describe constraints for the attribute value.

Error Information if property is missing/wrong value*

Here you MAY specify error information generated when this attribute is missing or wrong. This helps the user in proper debugging and it helps us in verifying the completeness and robustness of the software code.

Fields marked with (*) are optional.

If the element contains more then one attribute you MUST specify them all. Attributes are not numbered.

3.1.7. Other Information

This part MAY contain any other information that may be of help to the user.

3.2. Template for sample metadata.configuration file

This section presents the template for metadata configuration file description. You MUST use it while writing a guide to your own metadata configuration file.

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

--- Template beings here ---

1. General information

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

2. Introduction

`<introduction>`

3. Overview of the configuration file

`<file-overview>`

4. Examples

`<examples>`

5. Namespaces

`<namespace-prefix>=<namespace-URI>`

6. Schema elements explained

`<level>` **Element-Name:** `<namespace-prefix>:<element-name>`
Mandatory: `YES | NO`
Description: `<element-description>`
Possible values: `<element-values>`
Default value: `<element-default-value>`
Constraints (formatting, etc) on values: `<element-constraints>`
Error Information if property is missing/wrong value: `<element-error>`
Attribute: `<attribute-name>`
Mandatory: `YES | NO`
Description: `<attribute-description>`
Possible values: `<attribute-values>`

Default value: `<attribute-default-value>`
Constraints (formatting, etc) on values: `<attribute-constraints>`
Error Information if property is missing/wrong value: `<attribute-error>`

7. Other Information

```
<other-information>
```

--- Template ends here ---