

# PHP SQL-MA Service

### Document Change Log

Version Number	Date	Description of Change	People
1.0.0		First version issued	Murilo Vetter, Peterson Oliveira

## Table of Contents

1. General Information	pg 4
2. Installation	pg 4
3. Configuration	pg 4
4. Introduction	pg 5
5. Functional Specification	pg 5
1. Functionality handle-metadata	pg 5
2. Functionality handle-data	pg 6
3. Functionality handle-LS-interaction	pg 7
4. Functionality handle-echo	pg 7
6. Testing	pg 8
7. Messages Examples	pg 13

## 1. GENERAL INFORMATION

**Service Name:** sql-ma-php

**Service Type:** MA

**Version/release:** 1.0

**Service Description:** Measurement Archive Service

**Contact person(s):** Murilo Vetter

**Contact Information:** [murilo@pop-sc.rnp.br](mailto:murilo@pop-sc.rnp.br)

## 2. INSTALLATION

**Pre-requisites:** mysql-server, apache2, libapache2-mod-php5, php5, php5-mysql

To install the MA service, you will need only to decompress the file sql\_ma.tar.gz into the apache2 web directory, in Debian Linux systems usually it is defined in the directory /var/www/.

You will also need to set up the SQL database, which in this case it has been used MySQL.

## 3. CONFIGURATION

You will need to configurate the file, global.php which is in the folder sqlma/include, in that document you will find some variables that need to be set, like:

```
/** -----  
----  
SQL-MA Global Settings:  
This is the global configuration file to define the global settings.  
-----  
**/  
  
/** -- Global Settings -----  
---- **/  
define ("SERVER_ID", "");  

```

```

**/
define ("LOG_VERBOSITY", "");
define ("LOG_FILE", "");

/** -- Summarization Settings -----
----- **/
define ("DIR_BACKUP", "");

```

Also you have to configure the database, the structure is at the **perfsonar\_ma.sql** file, at **sqlma/db** directory, so you need to restore the structure using the following command:

```
mysql -u root -p -h localhost database example < ma db.sql
```

Finally it's the time to configure the crontab, just type `crontab -e` at the command line and copy the following lines into it.

```

*/5 * * * * /usr/bin/php /var/www/sqlma/lib/lookup_service/sync_ls.php >
/dev/null 2>&1
5 * * * * php /var/www/sqlma/lib/sum/summaryscript.php owdsummary 6
10 * * * * php /var/www/sqlma/lib/sum/summaryscript.php clmp_owamp 6
15 * * * * php /var/www/sqlma/lib/sum/summaryscript.php clmp_ping 6
55 * * * * php /var/www/sqlma/lib/sum/summaryscript.php clmp_bwctl 12

```

## 4. INTRODUCTION

This document details functionalities supported by SQL Measurement Archive service. Each section describes specific functionality as well as provides success and failures factors. It also lists message types implementing this functionality.

## 5. FUNCTIONAL SPECIFICATIONS

### 5.1. Functionality *handle-metatada*

#### 5.1.1 Description

This functionality allows processing metadata in order to determine the Key. Metadata, literally "data about data," is associated with monitored interface for purposes of its description. The Key is a pointer to the location of measurement data stored in relational database. The service takes metadata from the request and based on the content of metadata configuration file returns the Key along with whole metadata.

This functionality also allows processing `service.properties` file in order to determine some basic descriptive information about the service like name, description or type.

#### 5.1.2 Success Factors

The service returns the Key when metadata provided in the request is matched to some entry in the metadata configuration file.

### 5.1.3 Failure Factors

Metadata does not match with the configuration file, there are missing arguments or incorrect informations.

### 5.1.4 Interface Specification

MetadataKeyRequest

MetadataKeyResponse

## 5.2. Functionality *handle-data*

### 5.2.1 Description

This functionality allows providing the data that holds the information obtained from relational database as well as store new data. The service implementing this functionality locates data and retrieves stored data. To do so, it must issue a MeasurementArchiveRequest message which results in the delivery of MeasurementArchiveResponse message.

The contents of request message can consist of the metadata or Key, a combination of metadata and some filtering parameters or a combination of both. In addition to it, this functionality it provides a mean for the requestor to store measurement data provided as a part of the MeasurementArchiveStoreRequest message.

### 5.2.2 Success Factors

The service returns data either when a Key is provided or metadata is provided. The Key must correspond to existing measurement data stored in relational database while metadata must describe measurement data in the way it is described in metadata configuration file. When the MeasurementArchiveRequest applies some filter to the data start and/or end time for measurements must fall within the valid timeframe of collected data.

Measurement data storage requires MeasurementArchiveStoreRequest to contain either metadata to identify proper location to store data or directly a Key.

### 5.2.3 Failure Factors

When wrong Key is provided in the MeasurementArchiveRequest message, pointing to improper data location the service will return error. The service will also be unable to locate measurement data when metadata is not found in the metadata configuration file, or it is not properly configured

### 5.2.4 Interface Specification

SetupDataRequest

SetupDataResponse

MeasurementArchiveStoreRequest

MeasurementArchiveStoreResponse

MeasurementArchiveRequest

MeasurementArchiveResponse

### 5.3. Functionality *handle-LS-interaction*

#### 5.3.1 Description

This functionality allows a service to become known to other services. The service implementing this functionality registers itself to the Lookup Service with the use of LSRegisterRequest message. The content of this message consist of lookup information which details service parameters. Once registered, to communicate the changes of the lookup information, the SQL MA service issues LSRegisterRequest messages. We can also update the LS Information with a LSUpdateRequest and we can see if the service is available and update the status with a KeepAliveRequest. Also we can unregister the service with a DregisterRequest.

#### 5.3.2 Success Factors

The service registers itself provided that the metadata configuration file is filled and available as lookup information is taken from this source. The interaction with Lookup Service is possible provided that the Lookup Service to register with is up and running. Successful registration is confirmed with a key of data stored in Lookup Service contained in the LSRegisterResponse message.

#### 5.3.3 Failure Factors

Registration fails if Lookup Service is not available or the SQL MA service doesn't know Lookup Service access point stored in SQL MA service configuration.

#### 5.3.4 Interface Specification

LSRegisterRequest  
LSRegisterResponse  
KeepAliveRequest  
LSUpdateRequest  
DRegisterRequest

### 5.4. Functionality *handle-echo*

#### 5.4.1 Description

This functionality allows checking the health of the service. The service implementing this functionality simply responds back to a request with a success message proving that the web service is alive.

#### 5.4.2 Success Factors

The service responds with success EchoResponse message if the service is alive and the EchoRequest message is properly accepted by the service.

### 5.4.3 Failure Factors

EchoRequest fails if the service is not responding to requests or the internal processing doesn't work properly. If error is detected but the service is able to send response it may contain some error message to indicate the root of the problem.

### 5.4.4 Interface Specifications

EchoRequest

*EchoResponse*

## 6. Message Examples

### PING MeasurementArchiveStoreRequest

```
<nmwg:message id="message1_resp" type="MeasurementArchiveStoreRequest"
xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
xmlns:ping="http://ggf.org/ns/nmwg/tools/ping/2.0/"
xmlns:nmwgt="http://ggf.org/ns/nmwg/topology/2.0/">
  <nmwg:metadata id="meta1">
    <ping:subject id="subject">
      <nmwgt:endPointPair>
        <nmwgt:src type="ipv4" value="192.168.0.1"/>
        <nmwgt:dst type="ipv4" value="192.168.0.2"/>
      </nmwgt:endPointPair>
    </ping:subject>
    <ping:parameters id="parameters">
      <nmwg:parameter name="interval">10</nmwg:parameter>
      <nmwg:parameter name="count">10</nmwg:parameter>
      <nmwg:parameter name="deadline">103</nmwg:parameter>
      <nmwg:parameter name="packetSize">103</nmwg:parameter>
      <nmwg:parameter name="ttl">10</nmwg:parameter>
    </ping:parameters>
    <nmwg:eventType>http://ggf.org/ns/nmwg/tools/ping/2.0</nmwg:eventType>
  </nmwg:metadata>
  <nmwg:data id="data1" metadataIdRef="meta1">
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="1"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.521"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="2"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.551"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="3"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.545"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="4"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.605"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="5"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.717"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="6"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.578"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="7"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.590"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="8"
```



```

timeType="unix" timeValue="1259159871252" ttl="62" value="0.583"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="9"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.525"
valueUnits="ms"/>
    <ping:datum numBytes="64" numBytesUnits="bytes" seqNum="10"
timeType="unix" timeValue="1259159871252" ttl="62" value="0.551"
valueUnits="ms"/>
    </nmwg:data>
</nmwg:message>

```

## OWAMP MeasurementArchiveStoreRequest

```

<nmwg:message id="id01" type="MeasurementArchiveStoreRequest"
xmlns:cron="http://ggf.org/ns/nmwg/tools/cron/2.0/"
xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
xmlns:nmwgt="http://ggf.org/ns/nmwg/topology/2.0/"
xmlns:owamp="http://ggf.org/ns/nmwg/tools/owamp/2.0/">
    <nmwg:metadata id="meta1">
        <owamp:subject id="subject">
            <nmwgt:endPointPair>
                <nmwgt:src type="ipv4" value="192.168.1.1"/>
                <nmwgt:dst type="ipv4" value="192.168.1.2"/>
            </nmwgt:endPointPair>
        </owamp:subject>
    </nmwg:metadata>
    <nmwg:eventType>http://ggf.org/ns/nmwg/tools/owamp/2.0</nmwg:eventType>
    <owamp:parameters id="parameters">
        <nmwg:parameter name="direction">from</nmwg:parameter>
        <nmwg:parameter name="count">10</nmwg:parameter>
        <nmwg:parameter name="dscp">100</nmwg:parameter>
        <nmwg:parameter name="phb">10000</nmwg:parameter>
        <nmwg:parameter name="wait">1024</nmwg:parameter>
        <nmwg:parameter name="padding">50</nmwg:parameter>
        <nmwg:parameter name="delayStart">33</nmwg:parameter>
        <nmwg:parameter name="portRange">333</nmwg:parameter>
        <nmwg:parameter name="endDelay">1001-
1002</nmwg:parameter>
        <nmwg:parameter name="timeOut">1002</nmwg:parameter>
    </owamp:parameters>
    </nmwg:metadata>
    <nmwg:data id="data1" metadataIdRef="meta1">
        <owamp:datum receiveSynchronized="10" receiveTime="20"
receiveTimeError="30" sendSynchronized="40" sendTime="50" sendTimeError="60"
ttl="70" seqNum="80" timeType="90" timeValue="100"/>
        <owamp:datum receiveSynchronized="110" receiveTime="120"
receiveTimeError="130" sendSynchronized="140" sendTime="150" sendTimeError="160"
ttl="170" seqNum="180" timeType="190" timeValue="200"/>
    </nmwg:data>
</nmwg:message>

```

## OWAMP SUMMARY MeasurementArchiveStoreRequest

```

<nmwg:message id="localhost.-797874b8:127d2a718f6:-526b"
type="MeasurementArchiveStoreRequest"
xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">
    <nmwg:metadata id="meta1-1268077822939">
        <owampSummary:subject id="subject"
xmlns:owampSummary="http://ggf.org/ns/nmwg/tools/owamp/summary/2.0/">
            <nmwgt:endPointPair
xmlns:nmwgt="http://ggf.org/ns/nmwg/topology/2.0/">
                <nmwgt:src type="ipv4" value="192.168.0.1"/>
            </nmwgt:endPointPair>
        </owampSummary:subject>
    </nmwg:metadata>
</nmwg:message>

```

```

        <nmwgt:dst type="ipv4" value="192.168.0.2"/>
    </nmwgt:endPointPair>
</owampSummary:subject>
    <owampSummary:parameters id="parameters"
xmlns:owampSummary="http://ggf.org/ns/nmwg/tools/owamp/summary/2.0/">
        <nmwg:parameter name="direction">to</nmwg:parameter>
        <nmwg:parameter name="bucketWidth">0.000000001</nmwg:parameter>
        <nmwg:parameter name="count">10</nmwg:parameter>
    </owampSummary:parameters>

<nmwg:eventType>http://ggf.org/ns/nmwg/tools/owamp/summary/2.0</nmwg:eventType>
</nmwg:metadata>
    <nmwg:data id="data1" metadataIdRef="meta1-1268077822939">
        <owampSummary:datum duplicates="0" loss="0" maxError="2.01166E-6"
max_delay="0.064538" min_delay="0.0444832" sent="10" sync="true"
timeType="unix" timeValue="1270549893435"
xmlns:owampSummary="http://ggf.org/ns/nmwg/tools/owamp/summary/2.0/">
            <nmtn:time type="unix"
xmlns:nmtm="http://ggf.org/ns/nmwg/time/2.0/">
                <nmtn:start type="unix" value="14944504873784881099"/>
                <nmtn:end type="unix" value="14944504879717650669"/>
            </nmtn:time>
            <owampSummary:value_buckets>
                <owampSummary:value_bucket count="1" value="44492723"/>
                <owampSummary:value_bucket count="1" value="44502737"/>
                <owampSummary:value_bucket count="1" value="44483187"/>
                <owampSummary:value_bucket count="1" value="44508936"/>
                <owampSummary:value_bucket count="1" value="44631006"/>
                <owampSummary:value_bucket count="1" value="64538004"/>
                <owampSummary:value_bucket count="1" value="44748308"/>
                <owampSummary:value_bucket count="1" value="44581892"/>
                <owampSummary:value_bucket count="1" value="44577123"/>
                <owampSummary:value_bucket count="1" value="44602873"/>
            </owampSummary:value_buckets>
            <owampSummary:TTL_buckets>
                <owampSummary:TTL_bucket count="4" ttl="247"/>
                <owampSummary:TTL_bucket count="6" ttl="241"/>
            </owampSummary:TTL_buckets>
            <owampSummary:TTL_buckets>
                <owampSummary:TTL_bucket rorder="10" count="247"/>
            </owampSummary:TTL_buckets>
            <owampSummary:reordering_buckets>
                <owampSummary:reordering_bucket count="0" order="1"/>
                <owampSummary:reordering_bucket count="2" order="1"/>
            </owampSummary:reordering_buckets>
        </owampSummary:datum>
    </nmwg:data>
</nmwg:message>

```

## PING MeasurementRequest

```

<nmwg:message id="message1_resp" type="MeasurementRequest"
xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
xmlns:ping="http://ggf.org/ns/nmwg/tools/ping/2.0/"
xmlns:nmwgt="http://ggf.org/ns/nmwg/topology/2.0/">
    <nmwg:metadata id="meta1">
        <ping:subject id="subject">
            <nmwgt:endPointPair>
                <nmwgt:src type="ipv4" value="192.168.0.1"/>
                <nmwgt:dst type="ipv4" value="192.168.0.2"/>
            </nmwgt:endPointPair>
        </ping:subject>
    </nmwg:metadata>
</nmwg:message>

```

```

        <ping:parameters id="parameters">
            <nmgw:parameter name="interval">10</nmgw:parameter>
            <nmgw:parameter name="count">10</nmgw:parameter>
            <nmgw:parameter name="deadline">103</nmgw:parameter>
            <nmgw:parameter name="packetSize">103</nmgw:parameter>
            <nmgw:parameter name="ttl">10</nmgw:parameter>
        </ping:parameters>

<nmgw:eventType>http://ggf.org/ns/nmgw/tools/ping/2.0</nmgw:eventType>
    </nmgw:metadata>
    <nmgw:data id="data1" metadataIdRef="meta1"/>
</nmgw:message>

```

## PING SetupDataRequest

```

<nmgw:message id="message1" type="SetupDataRequest"
xmlns:select="http://ggf.org/ns/nmgw/ops/select/2.0/"
xmlns:nmgw="http://ggf.org/ns/nmgw/base/2.0/"
xmlns:nmgwt="http://ggf.org/ns/nmgw/topology/2.0/"
xmlns:ping="http://ggf.org/ns/nmgw/tools/ping/2.0/">
    <nmgw:metadata id="meta1">
        <ping:subject id="subject">
            <nmgwt:endPointPair>
                <nmgwt:src type="ipv4" value="192.168.0.1"/>
                <nmgwt:dst type="ipv4" value="192.168.0.2"/>
            </nmgwt:endPointPair>
        </ping:subject>
        <nmgw:eventType>http://ggf.org/ns/nmgw/tools/ping/2.0</nmgw:eventType>
        <ping:parameters id="parameters">
            <nmgw:parameter name="count">1</nmgw:parameter>
            <nmgw:parameter name="interval">2</nmgw:parameter>
            <nmgw:parameter name="deadline">3</nmgw:parameter>
            <nmgw:parameter name="packetSize">4</nmgw:parameter>
            <nmgw:parameter name="ttl">5</nmgw:parameter>
        </ping:parameters>
    </nmgw:metadata>
    <nmgw:metadata id="select">
        <select:subject id="subject" metadataIdRef="meta1"/>
        <nmgw:eventType>http://ggf.org/ns/nmgw/ops/select/2.0</nmgw:eventType>
        <select:parameters id="parameters">
            <nmgw:parameter
name="startTime">1279165910917</nmgw:parameter>
            <nmgw:parameter
name="endTime">1279165910997</nmgw:parameter>
        </select:parameters>
    </nmgw:metadata>
    <nmgw:data id="data1" metadataIdRef="select"/>
</nmgw:message>

```

## OWAMP\_SUMMARY SetupDataRequest

```

<nmgw:message id="message1" type="SetupDataRequest"
xmlns:select="http://ggf.org/ns/nmgw/ops/select/2.0/"
xmlns:nmgw="http://ggf.org/ns/nmgw/base/2.0/"
xmlns:nmgwt="http://ggf.org/ns/nmgw/topology/2.0/"
xmlns:owampSummary="http://ggf.org/ns/nmgw/tools/owamp/summary/2.0/">
    <nmgw:metadata id="meta1">
        <owampSummary:subject id="subject">
            <nmgwt:endPointPair>
                <nmgwt:src type="ipv4" value="192.168.0.1"/>
                <nmgwt:dst type="ipv4" value="192.168.0.2"/>
            </nmgwt:endPointPair>
        </owampSummary:subject>
    </nmgw:metadata>

```

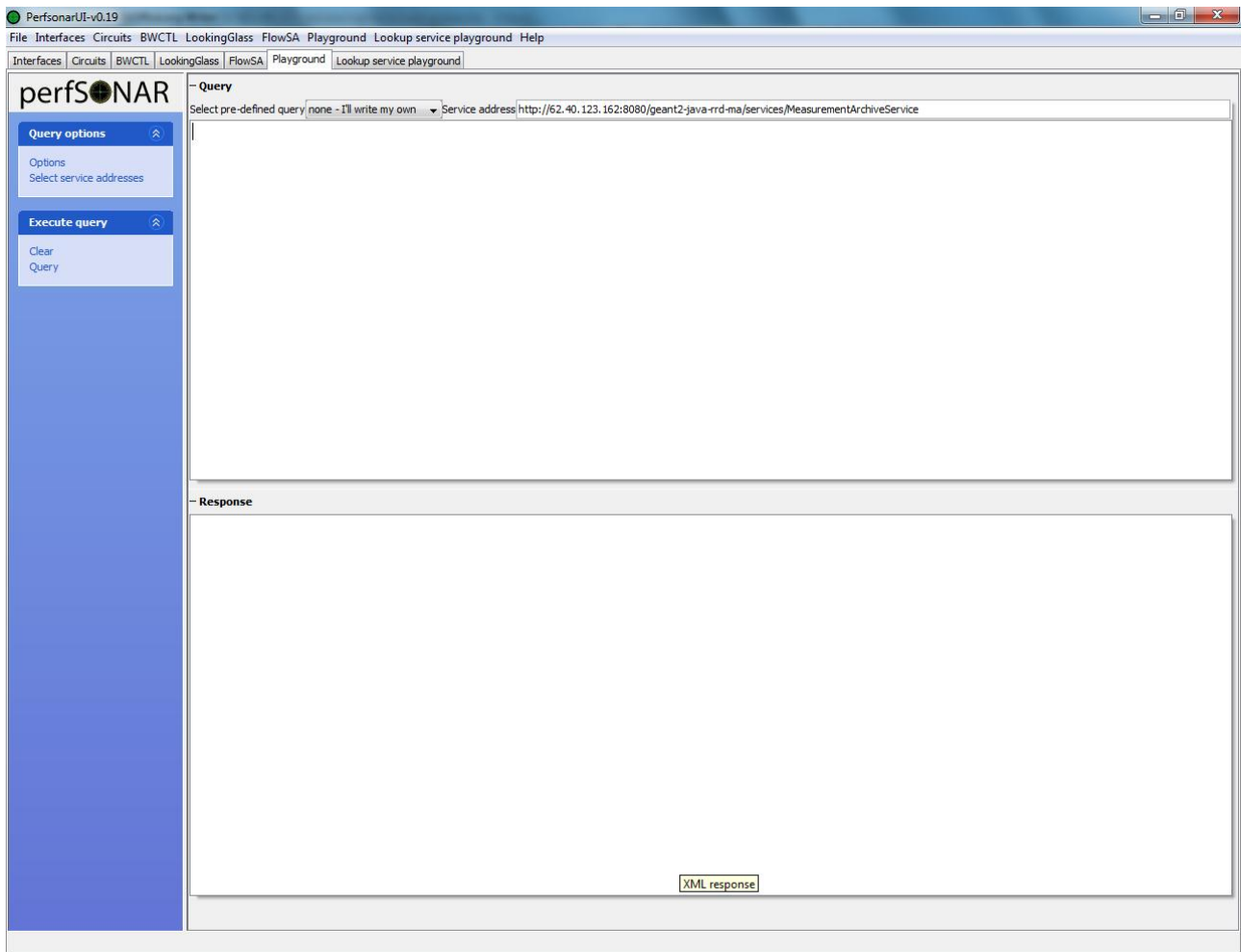
```
        </nmwgt:endPointPair>
    </owampSummary:subject>

<nmwg:eventType>http://ggf.org/ns/nmwg/tools/owamp/summary/2.0</nmwg:eventType>
    <owampSummary:parameters id="parameters">
        <nmwg:parameter name="count">1</nmwg:parameter>
        <nmwg:parameter name="interval">2</nmwg:parameter>
        <nmwg:parameter name="deadline">3</nmwg:parameter>
        <nmwg:parameter name="packetSize">4</nmwg:parameter>
        <nmwg:parameter name="ttl">5</nmwg:parameter>
    </owampSummary:parameters>
</nmwg:metadata>
<nmwg:metadata id="select">
    <select:subject id="subject" metadataIdRef="meta1"/>

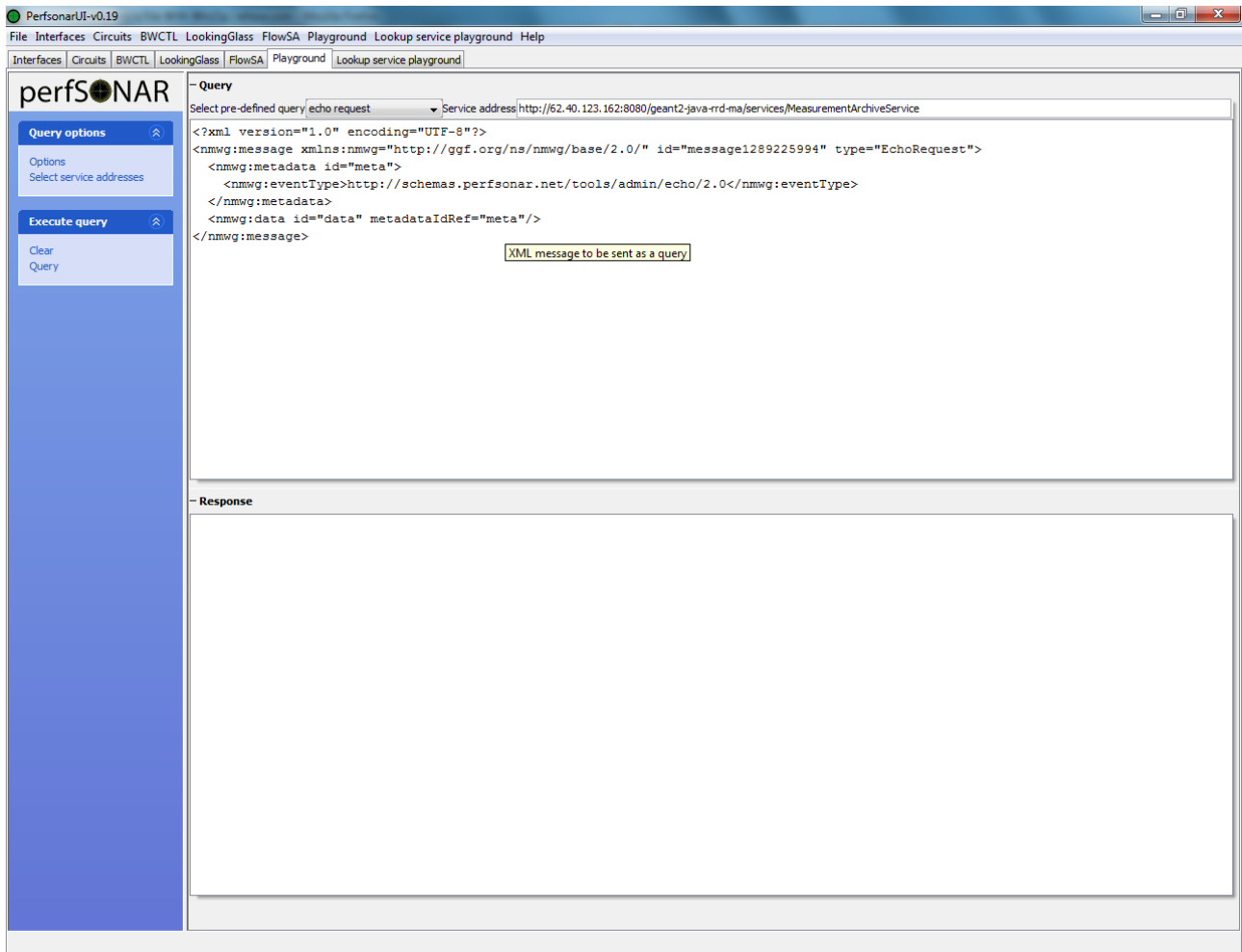
<nmwg:eventType>http://ggf.org/ns/nmwg/ops/select/2.0</nmwg:eventType>
    <select:parameters id="parameters">
        <nmwg:parameter
name="startTime">1279165910917</nmwg:parameter>
        <nmwg:parameter
name="endTime">1279165910997</nmwg:parameter>
    </select:parameters>
</nmwg:metadata>
    <nmwg:data id="data1" metadataIdRef="select"/>
</nmwg:message>
```

## 7. Testing

The tests can be done with the Perfsonar UI, which can be downloaded at the following URL  
“<http://www.perfsonar.net/download.html>”



Then to test your installation you will need to send an example message, it can be an echoRequest to the AP of your service.



The screenshot shows the perfSONAR web interface. The browser window title is "PerfsonarUI-v0.19". The menu bar includes "File", "Interfaces", "Circuits", "BWCTL", "LookingGlass", "FlowSA", "Playground", "Lookup service playground", and "Help". The main content area is titled "perfSONAR" and has a "Query" section. The "Query" section includes a dropdown menu for "Select pre-defined query" (set to "echo request") and a "Service address" field (set to "http://62.40.123.162:8080/geant2-java-rd-ma/services/MeasurementArchiveService"). Below this is a text area containing the following XML message:

```
<?xml version="1.0" encoding="UTF-8"?>
<nmwg:message xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/" id="message1289225994" type="EchoRequest">
  <nmwg:metadata id="meta">
    <nmwg:eventType>http://schemas.perfsonar.net/tools/admin/echo/2.0</nmwg:eventType>
  </nmwg:metadata>
  <nmwg:data id="data" metadataIdRef="meta"/>
</nmwg:message>
```

Below the XML message is a button labeled "XML message to be sent as a query". Below the query section is a "Response" section, which is currently empty.

After that you only need to execute the query, and wait for the service's response.

The screenshot displays the PerfsonarUI-v0.19 web interface. The main content area is divided into two sections: "Query" and "Response".

**Query Section:**

- Service address: `http://62.40.123.162:8080/geant2-java-rrd-ma/services/MeasurementArchiveService`
- Selected pre-defined query: `echo request`
- XML Query:

```
<?xml version="1.0" encoding="UTF-8"?>
<nmwg:message xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/" id="message1289225994" type="EchoRequest">
  <nmwg:metadata id="meta">
    <nmwg:eventType>http://schemas.perfsonar.net/tools/admin/echo/2.0</nmwg:eventType>
  </nmwg:metadata>
  <nmwg:data id="data" metadataIdRef="meta"/>
</nmwg:message>
```

**Response Section:**

- XML Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<nmwg:message id="message1289225994_resp"
  messageIdRef="message1289225994" type="EchoResponse" xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">
  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.echo</nmwg:eventType>
  </nmwg:metadata>
  <nmwg:data id="resultDescriptionData_for_resultCodeMetadata" metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">This is the echo response from the service.</nmwgr:datum>
  </nmwg:data>
</nmwg:message>
```

At the bottom of the response area, there is a button labeled "XML response".

The interface also includes a sidebar on the left with "Query options" and "Execute query" sections, and a status bar at the bottom left that says "Completed." with a green checkmark icon on the right.