

# XML Configuration Reader Manual

Jeff Landgraf (jml@bnl.gov)

## 1. Introduction

This is the new method for reading the STAR configuration files. In the old scheme, binary images of the configuration structure were produced every run and stored in the directory:

```
/RTS/conf/handler/cfg_04000
```

This file was guaranteed valid from the time that a node receives the "DAQ\_SEND\_CONFIG" message and the time that the node returns the "DAQ\_SEND\_CONFIG" message. The configuration file structure is defined in

```
cvs/StRoot/RTS/RC_Config.h
```

and is completely frozen. The original file was a fixed-size binary image, which made updating the system very difficult, as each online component had to be re-compiled for any change. The binary image is now obsolete, however for backward compatibility I continue to produce this file which contains all of the original configuration values as of Summer 2012.

However, any new parameters must be accessed using the new XML reader. The new scheme is identical, but the configuration information is stored in the file:

```
/RTS/conf/handler/cfg_04000.xml
```

## 2. The library

The reader code is located in:

```
cvs/online/RTS/src/XMLCFG
```

Linux compile:

- a. In the source file add the line:

```
#include "XMLCFG/SimpleXmlDoc.h"
```

- b. In the Make file include the lines:

```
include ../../../../StRoot/RTS/src/rtsmakefile.def
```

```
include ../../../../StRoot/RTS/src/rtsplusplus.def
```

and make sure that \$(STD\_DAQ\_OBJS) are included in the link specifications.

VX Works compile:

- a. In the source file add the line:

```
#include "XMLCFG/SimpleXmlDoc.h"
```

- b. In the startup file include the line:

```
ld < SimpleXmlLib
```

### 3. Reading the Configuration File

1. To read the configuration file follow these steps. One time only, run the command:

```
SimpleXmlDoc *xml = new SimpleXmlDoc();
```

2. At each "DAQ\_SEND\_CONFIG" command execute the commands:

```
char xmlname[100];  
getConfigBaseXml(xmlname, mptr);  
xml->close();  
xml->open(xmlname);
```

The first two lines are very similar to the protocol for the old configuration file. The `getConfigBaseXml()` function gets the name of the configuration file.

The second two lines read the file into local memory.

### 4. Reading Parameter Values

To read parameter values:

```
char *str = xml->getParam("paramName");  
int x = xml->getParamI("paramName");  
unsigned int u = xml->getParamU("paramName");  
float f = xml->getParamF("paramName");
```

The above data types are self explanatory. All fields are stored as strings in the data file, so the string version of the function always works. The typed versions attempt to do a conversion, and will return some appropriately typed value. The result is undefined if the parameter types do not match! There two special types used in the configuration file for bit masks that are displayed as check boxes in the configuration file. These are as follows:

```
int bit = xml->getParamBit(bit_idx, "paramName");  
unsigned int mask = xml->getParamMask(idx, "paramName");
```

The first tests whether a specific bit is set. The second gets the appropriate 32 bit component of the bitmask array for bitmasks with greater than 32 bits.

Because many parameters are array values, I also give a simple format for all of the `getParam()` functions that takes "printf" style specifiers. So

```
int x = xml->getParamI("trg_setup.triggers[%d].lx[%d].id", 5, 2);
```

is the same as:

```
int x = xml->getParamI("trg_setup.triggers[5].lx[2].id");
```

Which is the same as the obsolete:

```
x = cfg.trg_setup.triggers[5].lx[2].id;
```

## **5. The parameter list**

The list of available parameters can expand (or contract) at any time without affecting any unrelated system. The `getParam()` functions never fail. If a parameter is not defined by the run control, then the return value is an empty string ("") for character types, and "0" for all numeric types.

The history of the list of parameters available in the run control at any given time will be available at:

[http://www.star.bnl.gov/public/daq/rc\\_parameters.html](http://www.star.bnl.gov/public/daq/rc_parameters.html)