

# GEANT4/RADSRG Interface Manual

The RADSRG development team  
Lawrence Livermore National Laboratory

June 1, 2004  
Revised: June 2, 2004

## Abstract

RADSRG is a library for calculating gamma ray distributions. An initial material specification is aged and the daughter isotopes calculated to create the complete spectrum. RADSRG can be linked into, initialized, and called from other programs. This document specifies how to do this in GEANT4.

## 1 Configuring GEANT4 to use RADSRG

The user will need to have access to a GEANT4 installation. It is assumed that the user can already run a GEANT4 job.

Download the RADSRG library to your computer. It can be found at <http://nuclear.llnl.gov>. In the src directory type `gmake` in order to create the `libradsrc.a` file. *It is highly recommended that you determine what compiler the GEANT4 installation is using and modify the RADSRG Makefile to use the same compiler.*

The `g4` directory in the RADSRG release contains a sample GEANT4 job which accesses the `radsrc` library and runs a 10kg uranium ball problem. Type `source setup` to create the RADSRG environment variables needed by the Makefile and the code. If you have GEANT4 installed on your system and the GEANT4 environment variables are set then you should be able to type `gmake` to create an executable. An executable file called `exampleN01` should be located in the `bin` directory.

Type `bin/[system type]/exampleN01 example.in` to run the program.

The GEANT4 GNU makefile has been modified to link in the RADSRG libraries through the addition of two line:

```
EXTRALIBS += -L$(RADSRCHOME)/lib/ -lradsrg
CPPFLAGS += -I$(RADSRCHOME)/src/libradsrc/
```

The environment variable `RADSRCHOME` is defined in the `setup` routine along with the variable `RADSRG_LEGACYDATA` which points to the RADSRG data files.

## 2 Calling RADSRG from within GEANT4

The RADSRG routines are called from within the `ExN01PrimaryGeneratorAction` class which handles event generation for the problem. In the header files we include the RADSRG header files:

```
#include "radsrce.h"
#include "cpp_api.h"
```

and create some pointers variables that will point to instances of the RADSRC class,

```
radsrc::CRadSource* pRadSource;
radsrc::CRadSource* t1RadSource;
radsrc::CRadSource* t2RadSource;
```

the RADSRC routines live in the namespace `radsrc::` in order to prevent conflicts with geant4 classes.

To create a new instance of RADSRC one calls,

```
t1RadSource = radsrc::NewSource();
```

The instance of RADSRC must then be initialized with a problem definition. A number of ways to do that are provided. One way is to create a text file with the problem definition and pass the location of that file to the `LoadConfig` member function.

```
t1Good = radsrc::LoadConfig(t1RadSource, (const std::string) FileName);
```

If a NULL string is passed the program will look for the environment variable `RADSRCCONFIG` to find the input text file.

The text file must be in the form of,

```
U235      90.0
U238      10.0
AGE       20.0
```

where the fraction of each isotope is specified and should add up to 100%. To allow for problems with contamination the fractional sum can be slightly greater than 100%. The last line in the file should be an AGE card with the age given in years.

Additionally, the problem specification can be passed as input lines to the code,

```
radsrc::AddConfig(t2RadSource, ''U235 90.0'');
radsrc::AddConfig(t2RadSource, ''U238 10.0'');
radsrc::AddConfig(t2RadSource, ''AGE 20.0'');
t2Good = radsrc::SourceConfig(t2RadSource);
```

where the `AddConfig` member function accepts text lines of input and the `SourceConfig` processes the input and performs the RADSRC calculations and setup.

The RADSRC problem can also be specified from the GEANT4 command line through the commands defined in the `ExN01PrimaryGeneratorMessenger` class. They duplicate from the command line what is available in the code.

```
/radsrc/file './problem.in'
```

reads the problem definition from the specified file

```
/radsrc/file ''
```

will look for the environment variable `RADSRCCONFIG` to find the input text file. The complete problem specification can also be passed on the command line by using

```
/radsrc/input U235 90.0  
/radsrc/input U238 10.0  
/radsrc/input AGE 20.0  
/radsrc/update
```

The GEANT4 code is set up so that the `ExN01PrimaryGeneratorAction` class will try to find a RADSRC input text file at instantiation. A constructor function is provided which passes a string which specifies the file location. See `exampleN01.cc` for an example. That definition can be overridden from the command line. If no problem definition is specified the program will terminate at the first event.