SAND2016-8058

SANDIA REPORT

SAND2014-3933 REV1
UnlimitedRelease

GADRAS Isotope ID User's Manual for Analysis of Gamma-Ray Measurements and API for Linux and Android

Dean J. Mitchell and Lee T. Harding

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-3933 ZZZ
Unlimited Release
Printed June 8, 2015

GADRAS Isotope ID User's Manual for Analysis of Gamma-Ray Measurements and API for Linux and Android

Dean J. Mitchell and Lee T. Harding
Nuclear Threat Science, 6634
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185, MS-0782

Abstract

Isotope identification algorithms that are contained in the Gamma Detector Response and Analysis Software (GADRAS) can be used for real-time stationary measurement and search applications on platforms operating under Linux or Android operating systems. Since the background radiation can vary considerably due to variations in naturally-occurring radioactive materials (NORM), spectral algorithms can be substantially more sensitive to threat materials than search algorithms based strictly on count rate. Specific isotopes or interest can be designated for the search algorithm, which permits suppression of alarms for non-threatening sources, such as such as medical radionuclides. The same isotope identification algorithms that are used for search applications can also be used to process static measurements. The isotope identification algorithms follow the same protocols as those used by the Windows version of GADRAS, so files that are created under the Windows interface can be copied directly to processors on fielded sensors.

The analysis algorithms contain provisions for gain adjustment and energy linearization, which enables direct processing of spectra as they are recorded by multichannel analyzers. Gain compensation is performed by utilizing photopeaks in background spectra. Incorporation of this energy calibration tasks into the analysis algorithm also eliminates one of the more difficult challenges associated with development of radiation detection equipment.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

GADRAS	GAmma Detector Response and Analysis Software
HPGe	High Purity Germanium
API	Application Programming Interface

1. INTRODUCTION

This document describes the isotope identification approach that is employed by radiation analysis algorithms contained in the Gamma Detector Response and Analysis Software (GADRAS) and the API for operation on Linux and Android platforms. Three analysis paradigms currently exist: (1) Static and streaming search mode with hardware that provides accurate calibration or energy binned spectra (*calibrated mode*), (2) static and streaming search mode that relies on GADRAS's automatic calibration capabilities (*raw mode*) and (3) a file based mode that has the same functionality as DHSIsotopeID/HPGe isotope identification in the Windows version of GADRAS. All three methods are discussed in this document. R*aw mode* (discussed in Section 3.1) is a new addition to the IsotopeID Application Program Interface (API) starting with GADRAS Version 18.4.0. *Calibrated mode* (discussed in Section 3.2) has been part of the Linux API since GADRAS version 18.2.9, and the file based mode is discussed in Section 3.3.

The most significant change is the addition of analysis algorithms that are able to process "raw" gamma-ray spectra, where the adjective "raw" means that spectral data are passed to the analysis algorithm exactly as it was recorded by the multichannel analyzer without pre-processing. This approach alleviates the necessity for performing accurate energy calibration before passing spectra to the analysis algorithm. The analysis algorithms can accept data from multiple detector elements that are operated concurrently. Energy calibration is performed for individual elements before they are summed and processed as if the spectra were collected by a single large detector. The ability to analyze raw spectra does not preclude analysis of well calibrated or linearized spectra—in fact, analysis results are likely to be more reliable for stabilized detectors. Regardless of the quality of the input spectra, an advantage associated with adjustment of the energy calibration within the analysis module is that this approach ensures consistency between parameters that are applied to measured spectra and parameters that are used to compute the spectral templates used by the analysis algorithms.

The current versions of the analysis algorithms, which correspond to GADRAS Version 18.6.6 for Windows operating systems, eliminate the need to maintain different template database files for detectors that only differ as a result of small resolution variations. All of the GADRAS algorithms apply a full-spectrum analysis approach, where measurements are compared with combinations of computed templates that provide a best fit to the entire spectrum. The accuracy of the result degrades if the resolution of the measured spectrum is not consistent with the resolution that was applied when the spectral templates were generated. Resolution variations are accommodated by inspecting background spectra for peaks at 1460-keV and 2614-keV; the most suitable peak is used as a basis for estimation of the resolution parameter. This resolution is then used to interpolate the database to synthesize templates with the appropriate resolution. The default resolution is applied if a background spectrum is not provided or if the background does not exhibit features that are sufficiently distinctive to estimate the resolution.

The algorithms support analysis of either streaming search data or static measurements. Streaming search data consists of a continuous series of spectra that are passed to the algorithm upon successive calls. The algorithm maintains a moving background spectrum that, in addition to being used for gain compensation and resolution estimation, is also processed as input to the isotope identification algorithm. Foreground spectra are constructed from the most recent time interval and for sequences of up to eight consecutive intervals. This procedure provides flexibility to accommodate

differences in speed or source distances. Static measurements are processed in a similar way, except the background and foreground measurement are provided explicitly. The streaming search or static analysis algorithms can both accept data either from a single detector for a set of detectors that are operated concurrently from the same location. The algorithms return confidence levels that threat materials are present.

We also continue to support a file-based interface for the Linux API. The file based interface is analogous to the DHSIsotopeID or HPGeFSA analysis contained in the Windows version of GADRAS, and returns the same text result

2. ANALYSIS METHOD

2.1. Isotope ID Algorithms

The interface to the analysis algorithm is identical regardless of the type of detector that is used, but the approach to analyzing the data differs, depending on the resolution. The HPGeFSA algorithm is applied if the resolution is less than 2%[1], and the DHSIsotopeID algorithm is applied for detectors with poorer resolution.

2.1.1. DHSIsotopeID

The DHSIsotopeID analysis algorithm derives confidence estimates by analyzing the spectra using different combinations of isotopes. Solution sets are derived for naturally-occurring radioactive materials (NORM), and for NORM in combination with: industrial isotopes; medical isotopes; special nuclear materials (SNM); and combinations of all isotopes. Viable solutions may be obtained for different sets of radionuclides with spectral characteristics that cannot be resolved easily. For example, low-resolution detectors may not be able to distinguish radiation that is emitted by ^{131}I, ^{133}Ba, and ^{239}Pu, which all emit gamma rays in the 400-keV region. The assessment is especially ambiguous if spectral characteristics are obscured by the presence of materials that scatter the radiation. Although the solution may not be unique, the chi-square metric for the solution sets can be used to evaluate the relative probabilities. The estimated probabilities are used to generate a list of the most probable radionuclides, which are then submitted for analysis in a final solution set. Weighting factors are then recomputed for all of the solutions in order to assign a probability estimate for all of the isotopes in the database. The template database that is used to perform this assessment contains a variety of shielding configurations, which are interpolated to derive the best fit to the measured spectrum.

2.1.2. HPGeFSA

The HPGeFSA algorithm applies a full-spectrum analysis (FSA) approach that is essentially identical to the way a solution set is processed using DHSIsotopeID. However, photopeaks in high-resolution spectra can be inspected to derive a preliminary selection of radionuclides that are considerably more restrictive than can be obtained using low-resolution detectors. Selection of a set of isotopes based on low-confidence probabilities for the appearance of characteristic photopeaks enables HPGeFSA to perform only a single full-spectrum analysis solution, which is equivalent to the final solution set that is performed by DHSIsotopeID. The ability to jump quickly to a final solution is particularly beneficial for high-resolution detectors because spectra are evaluated using a greater number of energy groups, which would take longer to process otherwise. Consequently, spectra can generally be processed in a small fraction of a second on most modern processors regardless of the resolution.

One distinction between the ways in which HPGeFSA and DHSIsotopeID perform the FSA analysis concerns treatment of continuum regions. Spectra that are obtained by interpolation of precomputed templates do not reproduce some of the features in high resolution spectra, such as low-angle scattering below photopeaks, with the desired fidelity. Therefore, HPGeFSA introduces terms that accommodate differences between measured spectra and the computed templates. This

[1] The resolution is evaluated as the full-width at half height (FWHM) at 661 keV.

approach does not work well for spectra recorded with low-resolution detectors, nor is it necessary to do so since acceptable fits can generally be obtained using just computed templates. The tradeoff between the relative merits of two approaches occurs at about 2% resolution.

2.2. Streaming Search

2.2.1. Automatic Calibration in Raw Mode

The streaming search algorithm processes a continuous series of spectra that are passed as integer arrays. All data are treated as background spectra until completion of the first attempt at energy calibration. Because spectra are initially regarded as background, sensors that use this algorithm should be moved away from radiation sources upon startup. The algorithm estimates the time that is required to obtain a cumulative spectrum with sufficient statistical confidence to be suitable for energy calibration. The time varies inversely with the number of counts in the energy region associated with the photopeak that is used as a basis for energy calibration (see Section 2.4). The return value of the streaming search algorithm indicates the number of seconds remaining before energy calibration is attempted. The algorithm maintains a four-segment background buffer, and calibration is based on the sum of data in all of the background segments. Since the first calibration is attempted after filling only one background segment, the accuracy of the calibration improves after successive background segments are filled.[2] The algorithm applies the default energy calibration to analyze spectra until a successful calibration is achieved, so while the accuracy may not be optimal, an instrument that employs this algorithm can be used for search immediately after attempting the first energy calibration. The return value is set to zero if the calibration is successful; otherwise, the return value resets to the time remaining until the next calibration attempt. Error codes are defined in Section 3.1.1.

2.2.2. Rebinning of Calibrated Spectrum for Calibrated Mode

The *calibrated mode* streaming search method requires the spectra to be accurately calibrated or energy-binned before it can be streamed into analysis method. The simplest way of using the *calibrated mode* streaming search function would be to call the streaming search function and input energy bins and energy-binned data. Alternatively, you can use the spectrum rebinning function (Section 3.2.5), which will take the energy calibration, deviation pairs and energy bin structure provided by the caller, and rebin the spectrum into the specified energy group structure according to the energy calibration and the deviation pairs. This functionality was implemented primarily for use with sensors that comprise multiple detector elements, from which the outputs are summed and processed as if it were recorded by a single detector. The live-time and real-time parameters should be scaled accordingly (e.g., if you have 4 detectors with the same live time, the live time used for the analysis should be multiplied by 4).

[2] The oldest background segment is replaced after all of the background segments are filled so that the background spectrum is always current.

2.2.3. Streaming Search Analysis

The streaming search algorithm collects a running background spectrum with a duration that depends on the background count rate. The background duration is normally in the range 30 to 120 seconds. Analysis for the presence of radiation sources begins after initialization of the running background spectrum. Spectra that are passed to the search algorithm are added to the running background if no isotopes other than NORM are identified when the data are processed. If an isotope is present for more than two background periods, subsequent spectra are accrued as background. This procedure enables the identification of additional isotopes that may be brought into the vicinity of the detector after extended exposure to another source, but it also implies that the sensor will be relatively insensitive after extended exposure to a radioactive source.

Foreground spectra are retained in an eight-segment circular buffer, and the oldest segment is replaced upon successive calls to the search algorithm. Upon each call, the search algorithm composes four foreground spectra for possible analysis. The first foreground spectrum is the most recent spectrum that is passed to the algorithm; the second foreground is the sum of the two most recent spectra; the third foreground is the sum of the four most recent spectra; and the fourth foreground is the sum of the eight most recent spectra. These spectra are ranked for statistical differences relative to the current background based of the count rate, the chi-square difference, and ratios of count rates in broad energy regions. The foreground with the greatest statistical variation is submitted to the isotope identification algorithm if the variation is sufficient to suggest that it may not be the same as the current background. The current spectrum is added to the background buffer if the isotope identification algorithm determines that no isotopes other than NORM are present. The arguments in the call to the streaming search algorithm return a Stuff-Of-Interest (*SOI*) parameter, which is a floating point argument that scales the probability that something of interest is present. An *SOI* value exceeding 3 indicates reasonably high confidence that stuff-of-interest is present; this is recommended as an actionable value with high sensitivity and a low false alarm rate. The argument list also contains a string parameter *isotopeString*, which returns a text description of isotopes that were identified and the associated confidence values. Isotopes that are treated as SOI are defined by a file named "SOISpec.txt" in the application folder. The *SOISpec* string shown below is applied if file "SOISpec.txt" does not exist; spaces in the text string are significant because the analysis routines perform six-character comparisons of isotope names.

"Co60 Cs137 Ir192 U235 U238 Pu239 Am241 "

There is no latency in the identification isotopes if the analysis algorithm can identify a source based on data contained in a single time segment, but the identification may be delayed by up to eight seconds if additional time is required to perform the identification and the source is present for an extended period.

The identified isotopes are returned in the string parameter *isotopeString* with associated confidence values, and isotopes that are treated as SOI are defined by a file named "SOISpec.txt" in the application folder. The analysis can be reset and the running background can be cleared sending the initialization or reset signal using the *mode* variable.

2.3. Static Search for Calibrated Mode and Raw Mode

The static search algorithm is analogous to streaming search except the calling application specifies the foreground and background spectra. Static search is normally performed when a specific vehicle or container is inspected or as follow-up at a location where the streaming search algorithm produced an alarm. For the "raw mode", energy calibration is performed based on a specified photopeak in the background spectrum if a background is provided. The default calibration is applied if background is not available or if the background does not exhibit a characteristic peak with sufficient statistical confidence. The values of the arguments *SOI* and *isotopeString* are identical to arguments that are returned by the streaming search algorithm.

2.4. Energy Calibration Protocol for Raw Mode

The argument lists for the streaming search and the static search algorithms pass an array of values that specify the maximum energies (*eMax*) at the upper edges of the last energy group for each detector. If the spectrum is calibrated accurately and the lower edge of the first channel is zero, additional calibration is not required and analysis is performed using the asserted energy-group structure. The single character *calTag* parameter, which specifies how energy calibration adjustments are to be performed, can be set to a space character to inhibit gain adjustments so that *eMax* is applied without adjustment. However, most detectors exhibit uncompensated gain drift and nonlinearities, so the expectation of perfect energy calibration would be unusual. The GADRAS analysis algorithms can compensate for imperfect energy calibration as describe below.

2.4.1. Gain Adjustment

The parameter calTag defines how gain adjustments are performed. These adjustments are performed by adjusting the slope of the energy array such that an observed peak in the background spectrum is consistent the computed photopeak for a gamma ray of a defined energy. Setting calTag to "k" directs the energy algorithm to use the 40K peak at 1460 keV as a basis for gain adjustment; setting calTag to "t" directs the algorithms to use the location of the 2614-keV peak for gain adjustment. Calibration based on the 1460-keV peak is recommended because this is generally has more counts than the 2614-keV peak, particularly for small detectors. In order to function properly, background spectra must exhibit a clearly defined photopeak. Placing several grams of potassium near the detector in the form of potassium chloride or potassium carbonate can increase the count rate sufficiently to improve the reliability of the energy calibration procedure. Most digital multichannel analyzers (MCA) have stable offset, which do not require empirical adjustment. Although provisions exist in GADRAS for accommodation of offset drift, utilization of an MCA with stable offset is preferable.

2.5. Nonlinearities

Nonlinearities in the response of gamma-ray spectra are accommodated by defining "deviation pairs", which specify differences between the actual energies of the measured spectra versus energies derived from the calibration equation.[3] The methodology that is applied to the static and search algorithms restricts the calibration to a linear equation with zero offset at the lower edge of the first energy group, so the full-scale energy is the only adjustable parameter. Any deviation

[3] A cubic spline is used to interpolate for intermediate energies .

from this linear dependence is represented by the deviations pairs. The Windows version of GADRAS contains provisions to determining deviation pairs based on measurements of one or more calibration sources. The full-scale energy parameter should be adjusted such that the deviation is set to zero at the energy of the photopeak that will be used as a basis for gain adjustment. Figure 1 shows an example of a "Deviation.dat" file, which specifies a set of deviation pairs for each of two detectors, which are named "Aa1" and "Aa2". The detector naming convention corresponds to designators for radiation portals, and the only relevance to the search algorithm is that it provides a means for associating deviation pairs with specific detectors. When data for an array of detectors is processed, the first spectrum corresponds to "Aa1", the second spectrum is "Aa2", and this sequence continues through "Aa8". The sequence then resumes with "Ba1" through "Ba8", and so on. Deviations from linearity generally exhibit little temperature dependence, and they tend to be consistent over time. Therefore, deviation pairs that are derived from a one-time calibration of an instrument should be sufficient. Although detectors of the same type often exhibit similar nonlinearities, significant differences may exist.

```
Aa1
81      -5.00
122     -6.00
662    -12.00
898    -10.00
1461     0.00
1836    12.00
2615    23.00
!
Aa2
81      -6.00
122     -7.00
662    -14.00
898    -15.00
1461     0.00
1836     0.00
2615    -6.00
```

Figure 1. Content of the file "*Deviation.dat*", which defines a series of deviations as a function of energy (keV) for two detectors

Nonlinearities in the spectrum for the "*calibrated mode*" analysis method are handled through the rebinning function described in Section 3.2.5. The deviation pairs can be passed in as a matrix argument to the function.

3. APPLICATION PROGRAMMING INTERFACE (API)

As mentioned in the introduction, there are two methods of using the GADRAS Isotope ID. One is for calibrated spectra and the other is for raw spectra. Section 3.1 contains the methods for accessing the calibrated spectra functionality, Section 3.2 contains the methods for calling in to the raw spectra functionality, and Section 3.3 contains the method used to call the file based analysis mode. Both versions contain the static and streaming search functionality described in Sections 2.2 and 2.3, respectively.

3.1. Isotope ID for Raw Spectra

3.1.1. Error Codes and Return Values of Analysis Functions

The streaming search and the static search functions return integer values that are negative when errors are encountered. The return values for the error codes are listed below.

Table 1. Return values of the static and streaming search functions.

Return Value	Meaning
-2	Energy calibration failed
-1	An analysis error occurred or the template database file was not found.
0	Successful calibration and analysis.
>0	Time (seconds) remaining until next calibration (applies to StreamingSearch only)

An error code is returned if energy calibration failed for any of the detectors, which could be indicative of a detector malfunction, or it may just signify that the specified photopeak is too weak to be usable for calibration due to the local background conditions. While the significance of the error code is clear for a sensor that comprises a single detector element, the source of the problem may not be obvious when input data represents spectra that are recorded by several detectors. Therefore, the arguments include an integer array, *detStat*, which provides status information for individual detectors. Although the status for one or more detectors may indicate a failed energy calibration attempt, the analysis results may still be valid, particularly if accurate *eMax* values that are passed to the analysis algorithm.

Table 2. Values of the detector status arrays *detStat*.

Return Value	Meaning
0	Energy calibration was not performed (for example, calTag=" ")
1	Calibration was successful.
2	The specified background peak was not found.
3	There was a large error (based on chi-square) in the fit to the photopeak.
4	There were large uncertainties in the peak characteristics.

3.1.2. Detector Initialization

The following call initializes the application and detector settings that are used during subsequent calls to the search algorithm, and Table 3 defines the arguments.

status = InitializeIsotopeID(applicationFolder,detectorName,nChannels,nDetectors,calTag);

Table 3. Argument list for InitializeIsotopeID

Parameter	Type	Value
status	int32_t	Return values: -1 "Detector.dat" not found 0 success, 1 "Deviation.dat" not found, deviation pairs set to zero, 2 "SOIspec.txt" not found
applicationFolder	char*	Name of the application folder (e.g., /DHSIIDmobile).
detectorName	char*	Name of the folder containing the detector data (this is a sub-folder applicationFolder)
nChannels	int32_t	Number of channels per spectrum
nDetectors	int32_t	Number of detectors
calTag	char*	Energy calibration basis: " " Do not adjust the gain "k" Use 1460-keV peak for calibration "t" Use 2614-keV peak for calibration

3.1.3. Streaming Search

The *StreamingSearch* function is used to search for radiation sources by processing a continuous series of measured spectra. The calling convention is defined below and

Table 4 defines the arguments.

```
status = StreamingSearch (tl,tt,channelData,&SOI,&isotopeString,eMax, mode,
                          detStat,neutrons,&rateNotNorm);
```

Table 4. Argument list for *StreamingSearch*

Parameter	Type	Value
status	int32_t	Return value (see Table 1)
tl	array of size [nDetectors] float	Live time (s)
tt	array of size [nDetectors] float	Real time (s)
channelData	2-D int32_t array of [nDetectors, nChannels]	Spectra streamed from the calling application.
SOI	float*	Value that is scaled to the probability that "Stuff Of Interest" is present.
isotopeString	char**	String that lists all of the identified isotopes (all radionuclides, not just "stuff of interest"). The confidence level that these have been detected is in parentheses: (L) for Low, (F) for Fair, (H) for High. If nothing is detected, isotopeString will contain "NONE". The pointer is associated in Fortran, and must be freed by calling the function.
eMax	array of float[nDetectors]	Energy (keV) of the upper edge of the last channel.
detStat	array of int32_t [nDetectors]	Status of each detector (see Table 2)
mode	int32_t	Integer with values -1 for initialize, 0 analyze, 1 reset
neutrons	int32_t	Neutrons counts recorded during the current time interval
rateNotNorm	float*	Count rate that is associated with radionuclides that are not naturally-occurring isotopes

3.1.4. Static Search

After the detector is initialized, the *StaticSearch* function can be called to identify isotopes based on the analysis of measured foreground and background spectra. NORM peaks in the background spectrum are used for energy calibration and to derive the resolution parameters that are applied by the analysis algorithm. Hence, the background duration should be long enough to enable identification of the 1460-keV peak.[4] The calling convention is shown below and Table 3 defines the arguments.

[4] The foreground spectrum is analyzed using a synthetic background spectrum if a measured background is not provided, so the results may still be valid if eM*ax* is accurate. The value of the background live time parameter, TLB, must be set to zero if a measured background spectrum is not available.

```
status = StaticSearch(tlf, ttf, foregroundData, tlb, ttb, backgroundData, &SOI, &isotopeString, eMax,
    detStat, neutronsF, neutronsB, &rateNotNorm);
```

Table 5. Argument list for *StaticSearch*

Parameter	Type	Value
status	int32_t	Return value (see Table 1)
tlf	array of size [nDetectors] float	Foreground live time (s)
ttf	array of size [nDetectors] float	Foreground real time (s)
foregroundData	2-D int32_t array of [nDetectors, nChannels]	Foreground spectrum
tlb	array of size [nDetectors] float	Background live time (s)
ttb	array of size [nDetectors] float	Background real time (s)
backgroundData	2-D int32_t array of [nDetectors, nChannels]	Background spectrum
SOI	float*	Value that is scaled to the probability that "Stuff Of Interest" is present.
isotopeString	char**	String that lists all of the identified isotopes (all radionuclides, not just "stuff of interest"). The confidence level that these have been detected is in parentheses: (L) for Low, (F) for Fair, (H) for High. If nothing is detected isotopeString will contain "NONE". The pointer is associated in Fortran, and must be freed by calling the function.
eMax	array of float[nDetectors]	Energy (keV) of the upper edge of the last channel.
detStat	array of int32_t [nDetectors]	Status of each detector (see Table 2)
neutronsF	int32_t	Neutrons counts associated with the foreground collection time tlf
neutronsB	int32_t	Neutrons counts associated with the collection time tlb
rateNotNorm	float*	Count rate that is associated with radionuclides that are not naturally-occurring isotopes

3.2. Isotope ID for Calibrated Spectra

The interface for the Isotope ID algorithms in calibrated mode is defined by the c header file "dhsiid.h". The streaming search functionality is provided through the function SearchIsotopeID and the static search functionality is provided by StaticIsotopeID. These functions including a

function to re-bin spectra (*RebinSpectrumFrontend)*, and modifications of compatibility new function *SearchIsotopeID* were added to the GADRAS Isotope ID Linux libraries as of version 18.2.9. *SearchIsotopeID* while used in conjunction with *RebinSpectrumFrontend* provides the capability to re-bin spectra into a new energy-group structure. This functionality has been implemented primarily for use with spectra recorded by multiple detector elements to be summed so that they are processed as if they were recorded by a single detector element. Previous functionality provided by versions before 18.2.9 can be obtained through the use of SearchIsotopeID by setting the integer variable *newEcal* to 1 and providing updated energy calibration through the floating point array *ecal.*

3.2.1. Error Codes and Return Values of Analysis Functions

Table 6. Return values for *SearchIsotopeID* and StaticIsotopeID

Return Value	Meaning
0	Analysis was successful.
-1	An analysis error occurred or the template database file was not found.

3.2.2. Detector Initialization

The following call initializes the application and detector settings that are used during subsequent calls to the search algorithm.

```
status = initializedhsiidmobile(applicationFolder, detectorName, nchannels);
```

Table 7. Argument list for *initializedhsiidmobile.*

Parameter	Type	Value
status	int32_t	Return values (see Table 6)
applicationFolder	char*	name of the application folder (e.g., /DHSIIDmobile).
detectorName	char*	name of the folder containing the detector data (this is a sub-folder applicationFolder). The string must be padded
nchannels	int32_t	number of channels

3.2.3. Streaming Search

After the detector is initialized, the SearchIsotopeID function is called periodically during radiation search activities to provide real-time isotope identification based on the analysis of gamma-ray spectra. The calling convention is defined below and
Table 8 defines the arguments.

```
status = SearchIsotopeID(ecal,tl,tt,spectrumbuffer,&SOI,&IsotopeString,mode, newEcal,
    rebinnedEnergyGroups,neutrons,&rateNotNorm);
```

Table 8. Argument list for *SearchIsotopeID.*

Parameter	Type	Value
status	int32_t	return values 0 for success -1 for failure
tl	float	live time
tt	float	real time
spectrumbuffer	array of size nchannels, float	spectrum streamed from the calling application
SOI	float*	Value that is scaled to the probability of detection of specified "stuff of interest" that is specified in SOIspec.txt. Values of SOI of approximately 3 indicate a high confidence that "stuff of interest" is in the spectrum. Values of SOI between 0 and 2.5 indicate that something is likely in the spectrum
IsotopeString	char**	String which contains all of the identified isotopes from the analysis (includes anything identified, not just "stuff of interest"). The confidence level that these have been detected is in parentheses, as Low (L), Fair (F), High (H). Pointer is associated in Fortran, and must be freed by calling function
mode	int32_t	Integer with values -1 for initialize, 0 analyze, 1 reset
rebinnedEnergyGroups	array size of nchannels+1, float	Energy group structure defined by the caller. E[0] should be the lower edge of the first group, and E[nchannels] should be the upper edge of the last group.
neutrons	int32_t	Neutrons counts recorded during the current time interval
rateNotNorm	float*	Count rate that is associated with radionuclides that are not naturally-occurring isotopes

3.2.4. Static Search

The *StaticIsotopeID* function is analogous to *SearchIsotopeID*, except that it applies a background spectrum that is defined as an argument as opposed to the approach used by *SearchIsotopeID*, which maintains and applies a running background. Analysis is performed without use of the background if the background live time parameter, *tlb*, is equal to zero, but the *backgroundBuffer*

must still be passed as an argument. The calling convention is shown below and Table 9 defines the arguments.

```
status = StaticIsotopeID(tlf, ttf, foregroundBuffer, tlb, ttb, backgroundBuffer, &SOI, &IsotopeString,
    rebinnedEnergyGroups,neutronsF,neutronsB,&rateNotNorm);
```

Table 9. Argument list for *StaticIsotopeID*

Parameter	Type	Value
status	int32_t	Return values (see Table 6)
tlf	float	Foreground live time
ttf	float	Foreground real time
foregroundBuffer	array of size [nchannels], float	Foreground spectrum
tlb	float	Background live time
ttb	float	Background real time
backgroundBuffer	array of size [nchannels], float	Background spectrum
SOI	float*	Value that is scaled to the probability of detection of specified "stuff of interest"
IsotopeString	char**	String that lists all of the identified isotopes and an associated confidence level
rebinnedEnergyGroups	array size of [nchannels+1], *float*	Energy group structure defined by the caller
neutronsF	int32_t	Neutrons counts associated with the foreground collection time tlf
neutronsB	int32_t	Neutrons counts associated with the collection time tlb
rateNotNorm	float*	Count rate that is associated with radionuclides that are not naturally-occurring isotopes

3.2.5. Spectrum Re-binning

The *rebinnedEnergyGroups* function is used to re-bin spectra to an energy-group structure that is defined by the calling application. The function computes the input energy groups based on asserted energy calibration parameters and deviation pairs that are compatible with GADRAS. The array *rebinnedEnergyGroups* can define a linear energy group structure, but linearity is not a requirement. Energies should span a range of 0 to at least 3000 keV to make full utilization of spectral information, but compliance with this energy range is not a requirement. The calling convention is shown below and Table 10 defines the arguments.

RebinSpectrumFrontend(nchannels, spectrum, ecal, deviationpairs, rebinnedEnergyGroups, rebinnedSpectrum);

Table 10. Argument list for RebinSpectrumFrontend.

Parameter	Type	Value
nchannels	int32_t	number of channels in input spectrum and output spectrum
spectrum	Array size of [*nchannels*], *float*	Spectrum from the calling application
ecal	Array size of [4], *float*	Energy calibration coefficients in terms of FullRangeFraction
deviationPairs	Array size of [20][2] , *float*	Deviation pairs are used to modify the energy calibration and are specified in terms of energy and the deviation from that energy both values are in keV
rebinnedEnergyGroups	Array size of [*nchannels*+1], *float*	Energy group structure defined by the caller. E[0] is the lower edge of the first group, and E[nchannels+1] is the upper edge of the last group.
rebinnedSpectrum	Array of size [*nchannels*], *float*	RebinnedSpectrum is "spectrum" that has been re-binned into the energy group structure defined by *rebinnedEnergyGroups*

Energy boundaries (keV) associated with the spectrum that is passed as *spectrumBuffer* are computed according to the following polynomial equation:

$$E_i = c_0 + c_1 x_i + c_2 {x_i}^2 + c_3 {x_i}^3 \quad (1)$$

where *Ei* is the lower edge if channel *i*, and *x* is the fraction of the full energy range, and c_0 to c_4 correspond to the ecal array in the "calibrated mode" functions. Thus $x_l=0$ and $x_{n+1}=1$, where n is the number of channels. As described in Reference 1, the energy calibration array is adjusted based on deviation pairs if non-zero deviation are specified either in the file "Deviation.dat" or if non-zero deviation pairs are declared via calls to *RebinSpectrumFrontend*.

3.3. File Based Isotope Identification

Support for the file-based interface to the static isotope identification algorithm, which is provided by the function *hpgefsafrontend*, continues without change with the current release of GADRAS. Analysis is performed on demand and only requires one call, which handles all initialization and processing. The following procedure call invokes either HPGeFSA or DHSIsotopeID depending on the resolution of the detector, and Table 11 defines the arguments.

status = hpgefsafrontend(detectorName, analysisFile, eventSummary);

Table 11. Argument list for *hpgefsafrontend*

Parameter	Type	Value
status	int32_t	Return values, 0 for success -1 for failure
detectorName	char*	Name of the folder containing the detector data (this is a sub-folder of "./")
analysisFile	char*	Name of the pcf file containing either a single spectrum or foreground and a background pair. If a single spectrum is provided, a background will be synthesized
eventSummary	char**	Analysis result as a string that has identical format to the result that is returned from DHSIsotopeID/HPGeFSA in GADRAS.

3.4. Files Used by Isotope Identification Algorithms

In addition to the shared object files, the *ApplicationFolder* must also contain the following files:

CharacteristicGammas.txt:

This file lists characteristic gamma rays for radionuclides that are included in the template database. This file is distributed with GADRAS.

SOISpec.txt:

Isotopes that are declared to be Stuff-Of-Interest are specified in this file. The following is an example of a text string that declares seven isotopes to be Stuff-Of-Interest:

"Co60 Cs137 Ir192 U235 U238 Pu239 Am241 "

A folder specified by "detectorName" in the initialization functions is described by each analysis method (described in sections 3.1, 3.2 and 3.3) and must contain the following files, which describe characteristics that are associated with a specific detector.

DB.pcf:

This is a template database file that is created by GADRAS.

Detector.dat:

This file specifies detector response parameters.

Deviation.gadras (optional):

Deviation pairs that are listed in this file are used to linearize spectra for one or more detectors. The deviation pairs are set to zero if this file is not present.

DISTRIBUTION

1	MS0155	C. Hall	6634 (electronic copy)
1	MS0782	L. Harding	6634 (electronic copy)
1	MS0782	D. Mitchell	6634 (electronic copy)

[The housekeeping entries are required for all SAND reports.]

1	MS0899	Technical Library	9536 (electronic copy)


Sandia National Laboratories