

# HotSpot Software Quality Assurance Overview

Unclassified

September 12, 2019



## Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.

## Revision History

Revision Date	Originator(s)	Revision Description
12-1-2017	Fernando Aluzzi	Initial version. Applicable to HotSpot version 3.1
9-12-2019	Fernando Aluzzi	<ul style="list-style-type: none"><li>• Updated to version 3.1.1.</li><li>• Removed appendix section providing code illustration in .NET Framework</li><li>• Added new appendix section B addressing equivalence between Central Registry version and version 3.1.1</li></ul>

# Table of Contents

<b>Section 1.</b>	<b>Summary.....</b>	<b>v</b>
<b>Section 2.</b>	<b>Introduction.....</b>	<b>1</b>
<b>Section 3.</b>	<b>Version History Overview.....</b>	<b>4</b>
3.1	Release Numbering System.....	4
3.2	Minor Fixes.....	4
3.3	New Model Options/Enhancements.....	5
3.4	Microsoft Windows OS Requirements and Issues.....	5
3.4.1	Application Data Requirement.....	5
3.4.2	Other Windows Related Fixes.....	6
<b>Section 4.</b>	<b>Modernization and Migration to Microsoft Windows .NET Framework.....</b>	<b>7</b>
4.1	Background and Justification.....	7
4.2	Preliminary Migration Using the Microsoft Migration Wizard.....	8
4.2.1	Automated Line-to-Line Translations.....	8
4.2.2	Variable Type Changes.....	8
4.3	Compilation Errors Resulting from Wizard Migration.....	9
4.4	Post-Error Phase, Logic Issues.....	11
4.4.1	Event Handling.....	11
4.4.2	File Dialog Interfaces.....	11
4.5	Modernization beyond Migration Requirements.....	12
4.6	Migration Summary.....	13
<b>Section 5.</b>	<b>Usage and Central Registry Toolbox Justification.....</b>	<b>14</b>
5.1	Usage and User Base.....	14
5.2	Support Burden for Maintaining Central Registry Version 2.07.1.....	14
5.3	Upgrade of the 95 Percent Dose Methodology.....	14
5.4	Windows Application Data Update of Version 2.07.2.....	15
5.5	Migration/Modernization Upgrades.....	15
5.6	Consistency between Central Registry Toolbox Version 2.07.1 and the Latest Versions of HotSpot.....	15
5.7	SQA Procedures.....	16
<b>Section 6.</b>	<b>Description of Accompanying SQA Documentation.....</b>	<b>17</b>
<b>Appendix A.</b>	<b>HotSpot Release Notes.....</b>	<b>18</b>
HotSpot 2.07.2 – September 2011.....		18
HotSpot 3.0.0 – March 2013.....		18
HotSpot 3.0.1 – September 2013.....		19
HotSpot 3.0.2 – July 2014.....		19
HotSpot 3.0.3 – August 2015.....		20
HotSpot 3.1.0 – October 2017.....		21
HotSpot 3.1.1 – July 2019.....		22
<b>Appendix B.</b>	<b>Equivalence Verification between the Latest Version of HotSpot (3.1.1) and the Central Registry Version (2.07.1).....</b>	<b>24</b>
<b>Appendix C.</b>	<b>Key Terms and Glossary.....</b>	<b>33</b>

# Section 1

---

## *Summary*

---

There is a need to update the HotSpot Health Physics Codes (HotSpot), from version 2.07.1 to version 3.1.1, in the DOE Office of Environment, Health, Safety & Security Central Registry Toolbox (Central Registry). Six versions of HotSpot have been released since version 2.07.1 was added to the Central Registry in 2010, and the HotSpot Support and Development team have received many requests from DOE analysts to submit the most recent Hotspot version into the Central Registry.

Having the most current version of HotSpot, version 3.1.1, in the Central Registry will provide the following benefits to HotSpot users:

- A modernized version of HotSpot.
- New application features.
- A more robust version of HotSpot, due to a number of software bug fixes.
- Avoidance of a significant runtime problem with the Central Registry version of HotSpot in current Microsoft Windows Operating Systems.

The upgrades that have gone into HotSpot since version 2.07.1 are extensive. To maintain compatibility with its host Operating System (OS), Microsoft Windows (Windows), HotSpot was completely rewritten from an older 1990s era language that is no longer supported by Microsoft, as of 2008, to one that is up to date with current versions of Windows. Other modernization upgrades include improvements in graphics products, performance, error handling, and the ability to save user settings, as well as other improvements. HotSpot's deployment package now provides a modern Windows installer and a Digital Certificate for authentication. New application features include enhancements for modeling explosive releases and the ability to calculate percentile doses in a manner consistent with DOE-STD-3009<sup>1</sup>.

None of these modernization features or enhancements has affected HotSpot's core physics and methodologies. Therefore, HotSpot version 3.1.1 provides the same numerical results as version 2.07.1. This has been verified by Software Quality Assurance (SQA) testing procedures, established during the submission of 2.07.1 into the Central Registry, which have been strictly followed with each version release over this period.

Many DOE analysts are required to use HotSpot version 2.07.1 as its Central Registry presence alleviates a significant SQA effort required by their work. But the ability to run version 2.07.1 with Windows 7 and later versions is hindered due to an incompatibility between this HotSpot 2.07.1 version and the newer Windows systems. To circumvent this issue, a significant user support effort has been in place from both the HotSpot Support Team as well as personnel who directly support these users at their respective facilities. This effort to allow users of version 2.07.1 to run on current Windows systems has no guarantee

of continuing to succeed in the future especially with ever growing security policies for users of DOE computer systems.

This document summarizes the history and the modernization of HotSpot since version 2.07.1 for the purpose of establishing the validity and justification for updating the DOE Central Registry with HotSpot version 3.1.1, and it also complements the formal set of HotSpot SQA documents, the HotSpot Configuration Management Plan,<sup>2</sup> and the HotSpot Software Test Plan<sup>3</sup> that directly address HotSpot's continuing adherence to the SQA work activities as set forth by DOE O 414.1D<sup>4</sup> and DOE G 414.1-4.<sup>5</sup>

## Section 2

### *Introduction*

The HotSpot Health Physics Codes (HotSpot), originally developed by Steve Homann of Lawrence Livermore National Laboratory (LLNL), provides a first-order approximation of the radiation effects associated with the atmospheric release of radioactive materials.<sup>6</sup> HotSpot is currently developed and maintained by the National Atmospheric Advisory Center (NARAC), working closely with Mr. Homann. NARAC's support of HotSpot, which started in 2009, encompasses various phases such as code development, customer support, version control, bug reporting, and other activities.

Early in the collaboration, the HotSpot team, consisting of Mr. Homann and dedicated members of NARAC, developed new Software Quality Assurance (SQA) documentation and testing procedures, which eventually led to the acceptance of HotSpot version 2.07.1 into the DOE Office of Environment, Health, Safety & Security Central Registry Toolbox (Central Registry) in 2010. This version of HotSpot, 2.07.1, was developed prior to the collaboration. After the acceptance of version 2.07.1 into the Central Registry, the HotSpot team embarked on a software development phase, resulting in the releases of HotSpot shown in Table 1.

**Table 1. History of HotSpot Releases**

HotSpot Version	Month/Year Released	Description
2.07.1	March 2010	Central Registry Version
2.07.2	September 2011	New Explosion Parameters
3.0	March 2013	Modernized to a .NET Framework version
3.0.1	September 2013	Addressed Windows issues relating to 3.0 release and Windows 7 OS
3.0.2	July 2014	Software Bug fixes
3.0.3	August 2015	Software Bug fixes
3.1	October 2017	New historical meteorological data (95 <sup>th</sup> Percentile) features, Acute Dose upgrades and “User Friendly” Options
3.1.1	July 2019	Software Bug fixes

For each release, the HotSpot team has consistently followed the SQA practices documented during the evaluation of version 2.07.1 for acceptance into the Central Registry. These SQA procedures, such as validation testing, version control, bug reporting, just to name a few, have also remained consistent over this period. For example, the testing and validation process followed by the HotSpot team prior to releasing a new version remains unchanged between the Central Registry version of 2.07.1 and the latest release, version 3.1.1.

Section 2 of this document provides an overview of the HotSpot software development starting with version 2.07.2 and highlights important software bug fixes, the primary areas of feature enhancement are the HotSpot explosive release scenario and the post-processing of dose calculations using historical meteorological data, and the focus on Microsoft Windows (Windows) issues. The focus on Windows Operating System (OS) highlights the importance placed by the HotSpot development team in modernizing the application and maintaining compatibility with the ever-changing versions of Windows systems.

After version 2.07.2, a significant amount of HotSpot development focused on modernizing the programming language and development environment. For the purposes of clarity, the term “modernization” as it applies to HotSpot will now be refined to the following two interdependent terms:

- Migration – The conversion of the entire HotSpot code base from an older, unsupported Microsoft programming language to one of Microsoft’s .NET Framework languages.
- Modernization – The rewriting of a code segment in manner that maintains the logistical flow but uses newly available language features to improve robustness. Modernization can also mean a non-code upgrade such as a new installation program.

The extensive effort involved in the migration is described in detail in Section 3, which will also include the reasoning behind NARAC’s decision to support the code migration. The migration required numerous changes especially with the Graphical User Interfaces (GUI), graphical displays of numerical output, and event handling. The 2013 upgrade from HotSpot version 2.07.2 to version 3.0 was predominantly based on the .NET Framework migration.

While the .NET Framework migration required the modernization of many sections of the HotSpot code, the new environment still provided support for some of the older language constructs. One such example is the handling of errors; the newer .NET Framework language supports the modern method of error or exception handling yet still allows the option of using an older method. In such cases of having the option of maintaining the older method and still meeting the requirements of the migration, the HotSpot development team has worked toward removing all such older code constructs and taken advantage of the newer functionality of the .NET Framework. These changes are also described in Section 3.

Section 4 provides an overview of HotSpot’s usage and customer base along with a justification for updating the current Central Registry version with the most recent HotSpot release, version 3.1.1.

Brief overviews of the formal HotSpot SQA documentation, the HotSpot Configuration Management Plan<sup>2</sup> and the HotSpot Software Test Plan<sup>3</sup>, are provided in Section 5. These two documents describe the HotSpot SQA efforts in the work activities set forth in DOE O 414.1D<sup>4</sup> and DOE G 414.1-4.<sup>5</sup> The HotSpot Development and Support team have continued to follow the procedures described in these



documents since the inception of HotSpot version 2.07.1 into the Central Registry. This document serves to accompany and not replace those documents.

# Section 3

---

## *Version History Overview*

---

A summary of the key points and highlights of the HotSpot release history since version 2.07.2 are laid out in this section. For a complete list of the HotSpot release notes see Appendix A.

The individual software upgrades in all of the releases since 2.07.2 can be categorized as follows.

- Software bug fixes – either calculation-based or GUI related
- New model options/enhancements
- Microsoft Windows OS requirements/issues

Discussion of the code conversion or migration to the Windows .NET Framework, which served as the primary software upgrade in going from version 2.07.2 to 3.0, is reserved for Section 3 as it involved a very significant effort.

### **3.1 Release Numbering System**

The system for determining HotSpot's version numbers should first be explained. Prior to version 3.0 HotSpot did not follow a conventional version numbering system. Beginning with version 3.0 (or 3.0.0) the following convention has been in place.

The release number has three parts and each part is separated by a period (i.e., #.#.#).

- The first number indicates a major software upgrade (i.e., from **3.1.1** to **4.0**).
- The second number indicates a minor to intermediate software upgrade (i.e. from **3.0.3** to **3.1.0**).
- The third number indicates a version providing only software bug fixes (i.e., from **3.0.0** to **3.0.1**).

Any number that follows that which was increased will be reset to zero. When less than three numbers are shown, i.e. 3 or 3.1, zeros are implied in the remaining placeholders (3 = 3.0 = 3.0.0).

### **3.2 Minor Fixes**

The minor or software fixes to HotSpot after version 2.07.1 account for a large portion of the software changes over this period. In particular, there were two HotSpot capabilities necessitating the need for a majority of the software bug fixes:

- Reading and writing numerical data in an international format not used in the United States but commonly used outside of the United States
- Reading and processing of Global Positioning System (GPS) data

The international data format, or non-USA mode as it is called in HotSpot, refers to a Windows language setting using the comma character, “,”, as a decimal separator and a semi-colon character, “;”, as a list separator. Several errors arose from this capability even prior to the migration, but all such errors were eventually fixed using a new .NET Framework capability having no such counterpart in the older HotSpot language. As prevalent as these errors were through version 3.0.3, they did not affect any users within the U.S. as they were likely to use standard American language settings in their Windows environment.

The reading of Global Positioning System (GPS) data also required a significant number of minor fixes. The capability was improved to provide faster data processing and better error handling. This particular feature also relied on an older Windows Dynamic Link Library (DLL), which caused compatibility issues in the .NET Framework version. The post migration release of version 3.0.1 used a newly available .NET Framework library for reading GPS data, which enabled the removal of the older DLL from the deployment package. This GPS related development was a minor enhancement as well as a bug related fix and was not considered significant enough to warrant an increase in version numbers reflecting a new feature. This GPS capability is commonly used by DOE’s Radiological Assistance Program (RAP), ARG (Accident Response Group), and NTNF (National Technical Nuclear Forensics) teams.

### **3.3 New Model Options/Enhancements**

Version 2.07.2 introduced two new options into HotSpot’s explosive release scenario.

- The ability to model ballistic particles; material considered too large to be dispersed through the atmosphere following an explosion.
- A new additional method for calculating the overall height of an explosive cloud based on the Green Field Study.<sup>7</sup>

The upgrade from version 3.0.3 to 3.1.0 was based on several minor GUI enhancements and a significant upgrade to the statistics based 95<sup>th</sup> percentile capability. The core functionality of this capability was enhanced with new processing methods to allow Safety Basis analysts at DOE facilities to follow the DOE requirements set forth in DOE-STD-3009.<sup>1</sup>

A primary goal of the HotSpot development team is to implement as many “user-friendly” GUI options as are commonly found in modern software applications. Many of these GUI enhancements were implemented in the upgrade to version 3.1 and provide improved handling of user-specified settings, new color options, and persistence of those settings when a user returns to the application after exiting.

### **3.4 Microsoft Windows OS Requirements and Issues**

#### **3.4.1 Application Data Requirement**

Since the mid-2000s, Microsoft has designed its operating systems to accommodate multiple users. In accordance, modern Windows systems (since 2007) require that applications running on the Windows OS separate user specific settings from user-shared areas. Stated differently, a single application must be accessible by multiple users yet allow for user specific data and output. The latter is commonly known as application data.

The release of HotSpot version 2.07.2 provided this important separation. Older HotSpot versions, such as the Central Registry version, do not observe this requirement and will fail when executed on modern Windows systems. The ramifications of this issue, which continues to affect users who are required to use HotSpot 2.07.1 from the Central Registry, are discussed in more detail in Section 4.

### **3.4.2 Other Windows Related Fixes**

As discussed in Section 3, the upgrade to version 3.0 brought with it numerous changes to the handling of user interactions with the HotSpot GUI. The subsequent versions, 3.0.1 and 3.0.2, addressed minor Windows and GUI related issues that resulted from the modernization to version 3.0.

The version 3.0 upgrade also introduced a digital signature to the HotSpot deployment package. The code signing certificate validates the authenticity of the downloaded HotSpot installer on the user end.

The installer program was upgraded from a Microsoft supplied deployment program integrated with Visual Studio, the HotSpot Integrated Development Environment (IDE), to a commercially purchased product during the HotSpot version 3.0.3 upgrade. The installer upgrade allowed the HotSpot development team to have more control over the final HotSpot deployment package and to have the ability to avoid future changes on the part of Microsoft in their Visual Studio integrated installer.

## Section 4

---

### *Modernization and Migration to Microsoft Windows .NET Framework*

---

#### **4.1 Background and Justification**

Prior to version 3.0, which was released in 2013, HotSpot was written in Visual Basic 6 (VB6). VB6, which dates back to 1991, provided several unique programming features and also a very convenient method for building event based graphical interfaces. The last release of the VB6 integrated development environment (IDE) in 1998 pre-dates the current .NET Windows environment for code development. Up through version 2.07.2 (released in 2011), HotSpot development continued on the older VB6 IDE despite the fact that Microsoft had stopped supporting it in 2008.<sup>8</sup> Microsoft did continue to provide the necessary runtime and library files for applications developed in the VB6 IDE, but as far as the files executing properly on future versions of Windows, they only promised an "It Just Works" commitment. Therefore, continuing to use the VB6 IDE in newer versions of Windows was risky.

Another reason for modernizing HotSpot was that the Windows OS in which the VB6 IDE had been supported, Windows XP, which Microsoft released in 2001, was soon to have its support terminated as well (support officially ended in 2014).<sup>9</sup> Continuing to use VB6 as the HotSpot IDE would therefore require maintaining a system running the unsupported Windows XP OS, a practice that is not consistent with security issues and computer support at DOE facilities.

With the termination of IDE support and updates for VB6 in 2008 as well as the pending termination of support for Windows XP, the HotSpot code base needed to be modernized to the .NET Framework in order to:

- Continue developing HotSpot in a Microsoft supported IDE
- Be able to update the IDE with version upgrades and bug fix patches
- Take advantage of new language features provided by the .NET Framework such as exception handling, inheritance, XML, and more
- Develop and build the HotSpot code in the same environment that most users run HotSpot: Windows 7 and later versions

After the release of HotSpot version 2.07.2, NARAC made the decision to support the migration to the .NET framework, by converting HotSpot's development language from VB6 to the .NET Framework version of Visual Basic, Visual Basic.NET (VB.NET). NARAC decided to declare this migration to VB.NET a major software upgrade from version 2.07.2 to a version 3.0.0. There were also some minor non-migration upgrades to version 3.0.0, which are described in the Release Notes (Appendix A).

New releases of the Windows operating system also brought about required updates to HotSpot such as the application data issue described in Section 2.4.1. HotSpot development that addresses such requirements is independent of the modernization discussed here.

## 4.2 Preliminary Migration Using the Microsoft Migration Wizard

The first step in converting the VB6 version of the code took advantage of the Visual Basic Upgrade Wizard (Wizard), a Microsoft tool included in the Visual Studio 2008 IDE and designed for this purpose.

### 4.2.1 Automated Line-to-Line Translations

While the migration Wizard can translate a simple VB6 program and automatically provide the same GUI functionality and output in VB.NET, the HotSpot code base was too complex for that to happen. The wizard was designed to work only with a single line of code for which a direct translation was available. However, the wizard was not designed to analyze a code segment (two or more consecutive or related lines) because logic paths are not easily determined. The following list provides a summary of the code translations that the Microsoft migration Wizard was able to provide.

- Conversion of the lower bounding index of all arrays from 1 to 0.
- All functions designated as event handlers had their signatures redefined accordingly to the .NET format.
- Special system calls that had new representations in VB.NET were changed accordingly.
- VB.NET supplied a VB6 wrapper Library, allowing for certain VB6 functionality to be preserved. VB6 functions having wrapper-defined counterparts were automatically converted.
- Since the list of keywords was expanded from VB6 to VB.NET, any variable in VB6 whose name matched a new keyword in VB.NET was renamed to avoid conflicts.
- Other simple one-line statements whose syntax had a direct line-to-line translation such as Control Structures (i.e., *while* loops and *for* loops) were converted.
- The entire code base was reorganized into an Object-Oriented structure.
  - All GUI windows (*Forms* in VB6) were redefined as VB.NET *classes* and all of the variables and functions associated with the window became members of that *class*.
- The concept of a *Module* was carried over from VB6 code to VB.NET. All of the Modules in the VB6 version of HotSpot were redefined in the VB.NET version.

The overall design and code structure were maintained while converting from the VB6 version to the Wizard generated VB.NET version.

### 4.2.2 Variable Type Changes

Some differences exist between the available variable types in VB6 and VB.NET. For example, VB6 has only three types of integers: byte, integer, and long, whereas VB.NET expands this list with the addition of five other integer types. Table 2 describes all the differences in variable types between VB6 and VB.NET and the corresponding changes made by the wizard.

**Table 2. Variable Type Conversions between Visual Basic 6 and Visual Basic .NET**

VB6 Variable Type	VB.NET Variable Type	Wizard Modification
Integer (2 bytes)	Short (2 bytes)	Size of variables declared as type Integer in VB6 preserved to be 2 bytes. Re-declared all 2-byte integers as type <i>Short</i>
Long (4-byte integer)	Integer (Int32, 4 bytes)	Size of variables declared as <i>Long</i> in VB6 preserved to be 4 bytes. Re-declared all 4-byte integers as type <i>Integer</i>
Undeclared types	Re-declared as Type Object (4 bit on 32-bit systems, 8 bit on 64-bit systems)	Unlike VB6, VB.NET does not allow variables to be declared without a type. Each undeclared type re-declared as an Object.
Variant	Does Not Exist	Variant variable type represents a variable whose type and size can be redefined during runtime. Such a variable type does not exist in VB.NET. Wizard action same as for undeclared types.

The following VB6 variable types were unchanged and map directly to a corresponding VB.NET variable:

- Boolean (size is platform dependent)
- Byte (unsigned integers from 0 to 255, 1 byte)
- Single (float, 4 bytes)
- Double (float, 4 bytes)
- Date (date and time structure, 8 bytes)
- Object – Used in VB6 and VB.NET to indicate a generic variable

Each variable conversion preserved the memory size of the original variable. This provides consistency in the precision of calculated results between VB6 versions of HotSpot (2.07.2 and earlier) and VB.NET versions (3.0 and later).

### 4.3 Compilation Errors Resulting from Wizard Migration

Although the Wizard provided a useful starting point to the migration, the resulting output contained approximately 5,000 errors hence not allowing HotSpot to compile. These errors were the result of two main reasons:

- The referenced VB6 statement was not available in VB.NET. Examples included special system calls and properties of VB6 GUI components. In some cases, the functionality was no longer being supported, it had been moved, or was implemented in a different manner.

- A given line of code could not be translated because it was part of a block of interdependent statements and the wizard was designed to make no assumptions in these situations as to that interdependency.

Many of these errors were due to VB.NET being an object-oriented language whereas VB6 is not. VB.NET requires more of the software developer in terms of responsibility. For example, VB.NET development must handle GUI component instantiation and destruction with functions such as *New* and *Dispose*, both common object-oriented functions. VB6 on the other hand, handled as much as possible behind the scenes on behalf of the developer. In essence, VB6 required much less from a software developer to be able to create *event-driven* GUI applications. On the other hand, creating *event-driven* GUI applications in VB.NET requires much more from the developer in terms of Object-Oriented principles and *event* details.

The following list provides examples of VB6 items or properties that were not carried over to VB.NET by the migration Wizard hence generating a significant number of errors requiring fixes:

- Use of Microsoft Powerpacks
- Printing
- Opening and closing application forms/windows
- Cancelling out of a file dialog
- Window resizing
- Help application
- Deprecated VB6 features
- Graphics – Anything related to drawing an object or text on a Form or Window

Graphics routines deserve special mention as that entire methodology was completely modified from VB6 to VB.NET resulting in errors from all graphic references (drawing shapes, writing freeform text on GUIs, handling images within forms, etc.). VB6 handled the details of graphics calls on behalf of the developer whereas VB.NET requires the developer to define and handle all graphics related code such as:

- Defining *Graphics* objects prior to drawing
- Handling redrawing (visual persistence) when forms or windows are moved or resized
- Properly disposing of Graphics objects when drawing is completed

HotSpot generates several graphical output products directly, without using third-party software, such as:

- Contour Plots on the Plume Centerline
- Contour Plots in a Compass based background
- Plume Centerline Plots
- HotSpot Mapping

The change in Graphics methodology between VB6 and VB.NET required the majority of the development effort in this phase of the migration. An example of a typical graphics change is provided in Appendix D.



## 4.4 Post-Error Phase, Logic Issues

Once all the compilation errors were removed and the code could now be built and executed, an extensive testing period began. During the testing, behavioral differences were observed between the VB6 and VB.NET versions primarily in the areas of event handling and file dialog interfaces. Both of these are discussed next.

### 4.4.1 Event Handling

Both VB6 and VB.NET are event driven (i.e. code blocks that can be associated with a specific GUI event such as a button click, mouse movement, or window resizing, etc.). During testing it was discovered that important sections of event driven code were being triggered in a different order between the VB6 and VB.NET versions. These situations are categorized as follows:

***The order of a Resize and Load (for a Form) event are reversed between VB6 and VB.NET***

The opening of a form or window is identified as a *load* event in both the VB6 and VB.NET languages. By comparison, a *resize* event is triggered not only when a user resizes a window, but also when a form first loads. The action of opening a form is therefore associated with both of these events. The order of these two events was reversed between the VB6 and VB.NET languages. This had a major impact on the initial VB.NET version of HotSpot since both events were used to call their own specific functions. To match the overall behavior of the VB6 version, many code blocks were moved and re-ordered in the VB.NET version.

***The event handling functions for two specific GUI components, RadioButtons and CheckBoxes, were being executed earlier than expected***

For example, HotSpot saves the status of all of its GUI parameters whenever a user exits the program. That information is then used when a user re-enters the program to set the same GUI values upon the user's return. During program start-up, code statements that set the state (checked or unchecked) of these components were also triggering the component's event code; the code intended to run when a user themselves selects or de-selects the component with the mouse.

The VB6 version of HotSpot did exhibit some of this behavior as well but not to the same extent. The VB.NET version of HotSpot was modified to not have event code for these components trigger where it was not intended.

### 4.4.2 File Dialog Interfaces

Another difference between VB6 and VB.NET was File Dialog Interfaces, a common interface for saving or opening files. In VB6 a single File Dialog entity could be defined and used for both opening and saving files to disk. This duality in the behavior of an interface is a good example of the many conveniences provided by VB6 that were no longer available in VB.NET.

During the testing phase, it was discovered that the Wizard did not handle File Dialogs in a robust way. In all areas requiring a File Dialog, the wizard made a guess whether to create a File Dialog for opening a file or one for saving files. Several of these sections resulted in the incorrect type of File Dialog for the intended purpose. In all these cases, the existence of the correct File Dialog was verified and restored if needed.

As listed in the next section, this opportunity was used to completely modernize all File Dialog interactions to improve the handling of user-initiated File Dialog interactions.

#### 4.5 Modernization beyond Migration Requirements

Since developing the first VB.NET version, the HotSpot development team has been able to identify sections of code that could be modernized, even though not required, by the initial migration to the .NET Framework. One example is the backward compatibility in VB.NET for some types of VB6 functionality. The identification of code sections in need of modernization has been based on a best practices approach toward Object-Oriented software development as well as the advantageous use of newly available features in VB.NET.

As time permits, each release of HotSpot version after 3.0 has included modernization to selected sections of code. Arguably, these upgrades were not considered as either software bug fixes or new features and hence are not usually detailed in a version's release notes. The following list describes the most common of these upgrades.

- Error/Exception Handling – Migrating away from jumping to a different section of code for handling an error by using a “go to” based statement toward the “Try/Catch” concept that is commonly used in modern day languages. VB.NET allows for both constructs.
- File Dialogs – All code using file dialogs now handle all possible actions that nullify the file operation (e.g., when a user cancels or aborts the interface).
- Validation – The code now takes advantage of IDE validation capabilities to perform error checking on user entered data within a GUI.
- File Access Conflicts – Eliminating the possibility of a file being opened simultaneously for reading or writing by more than one process.
- Reading files – Migrating toward taking advantage of newer VB.NET functions for reading and parsing text and away from processing text on a character-to-character basis.
- User settings – Migrating away from writing custom files to save user settings and toward using settings functionality within the VB.NET IDE.
- Object Oriented Concepts:
  - Modules and global variables – Although object oriented in nature, VB.NET still allows for the non-object-oriented concept of global variables by allowing the use of modules. The use of variables declared in modules with global access is being phased out.
  - Inheritance – The use of inheritance was first introduced in version 3.0.3 and is expected to grow significantly in future versions as it provides reusable code and facilitates bug fixes.
  - Structures – Though still allowed in VB.NET, these are being replaced with Classes.

While some of these concepts have been applied everywhere in the code, others continue to be addressed. There have also been cases where modernization upgrades served to correct a software bug as mentioned in Section 2.2; the international language settings and GPS based software fixes are examples.

#### **4.6 Migration Summary**

The migration Wizard provided a preliminary, although non-working, version of the application by addressing much of the syntax changes from VB6 to VB.NET format. The post-wizard phase focused on numerous GUI, graphics, and event handling issues. Once a working version was reached, unforeseen issues arose due to changes in event handling between the old and new languages. After taking care of those issues, extensive testing eventually showed that HotSpot output in VB.NET (versions 3.0 and later) consistently matched that of the VB6 versions (2.07.2 and earlier).

The migration effort focused on attaining a working version of HotSpot within the .NET Framework environment as quickly and efficiently as possible while maintaining the codes logistical flow and preserving its core numerical calculations. More simply stated, the primary guideline was to attain a VB.NET version that looked and behaved like and provided the same results as the VB6 version. The decision to migrate from Visual Basic 6 into the Visual Basic counterpart within the .NET Framework facilitated this effort as areas of backward compatibility was provided by the latter. The overall code logic and structure was preserved as much as possible to expedite the development of a final working version which could be deployed to the user community. A complete rewrite of the code into a truly object-oriented paradigm or into a different language was considered too expensive and impractical at that point in time. This could be addressed again in the future.

## Section 5

---

### *Usage and Central Registry Toolbox Justification*

---

#### **5.1 Usage and User Base**

HotSpot was originally developed to provide a range of analysis tools for various radiological safety needs within DOE/NNSA. HotSpot has numerous users, approximately three to five thousand overall, based on user downloads. Although HotSpot was originally developed for DOE/NNSA use, there are now many users outside of DOE/NNSA such as DHS, FBI, CDC, EPA, FEMA, USDA, local government, the military, fire departments and hazardous material teams, academia, private corporations, and similar organizations outside of the U.S.

It is estimated that about 25 to 35 percent of the user base works on behalf of DOE/NNSA, many of which are analysts with the responsibility for preparing Safety Basis or Emergency Planning documents, both of which require SQA documentation to assure the quality of their published results.

When these analysts choose to use HotSpot, they are faced with the dilemma of wanting to take advantage of the features provided by newer releases of HotSpot, but on the other hand satisfying the requirement of SQA documentation by using the older HotSpot version 2.07.1. For those choosing the latter option, NARAC has always provided and supported two downloadable versions of HotSpot: the most current version as well as the Central Registry version of 2.07.1. Illustrating the significance of the need to use the older Central Registry version, there have been over 3,000 downloads of version 2.07.1 since 2010.

The HotSpot Support and Development team has received many requests from both Safety Basis and Emergency Planning analysts to submit the most recent version of HotSpot into the Central Registry. While these users only represent approximately 15 to 25% of the overall HotSpot user base, they do represent a significant fraction of the HotSpot user base within DOE. Having the most current version of HotSpot in the Central Registry would provide several benefits as described below.

#### **5.2 Support Burden for Maintaining Central Registry Version 2.07.1**

Having the latest version of HotSpot in the Central Registry would alleviate the burden on the HotSpot support team of hosting two versions as well as the support provided directly to the users when running version 2.07.1 on newer Windows systems. The HotSpot support team consistently assists users of the Central Registry version due to the user not having Windows application data upgrade as described both below and in Section 2.

#### **5.3 Upgrade of the 95 Percent Dose Methodology**

As requested by Safety Basis analysts, version 3.1 introduced a new capability within the utility for using actual meteorological data from DOE on-site towers in the calculation of 95<sup>th</sup> percentile dose calculations. Based on DOE-STD-3009<sup>1</sup> and NRC Guide 1.145<sup>10</sup> Position 3, this feature calculates distances from an

atmospheric release point within a DOE/NNSA facility out to the site boundary along the standard sixteen wind direction compass directions. While these distances can be calculated manually from output results in previous versions of HotSpot, doing so requires a significant amount of post-processing normally done in a spreadsheet format. The ability for a Safety Basis analyst to use this feature would not only save this post-processing time, but also eliminate the possibility of user error.

#### **5.4 Windows Application Data Update of Version 2.07.2**

The current Central Registry version of 2.07.1 contains no enhancements to maintain compatibility with changes in the Windows OS. As discussed in Section 2, HotSpot version 2.07.2 provided the important update to use a proper, Windows allowable, location for writing and reading application data files. That update removed an important incompatibility affecting typical users of the Central Registry version of HotSpot on post Windows XP systems such as Windows 7.

To run the Central Registry version of HotSpot on current Windows systems, users must either:

- Maintain HotSpot on an older Windows operating system such as XP or earlier or
- Run HotSpot with elevated privileges to circumvent the application data requirement

The first of these options has become nearly impossible because Windows XP is no longer supported by Microsoft and for this reason as well as security reasons, DOE/NNSA sites have migrated to newer versions of Windows. The second option of running with elevated privileges is normally the chosen option among users of the Central Registry version of HotSpot, 2.07.1, but this workaround is a support burden on both the corresponding computer system administrators at these facilities as well as the HotSpot support team. Maintaining special privileges for HotSpot users, given increasing security requirements at the DOE facilities, is an inefficient solution, which does not promise to last indefinitely.

#### **5.5 Migration/Modernization Upgrades**

Another benefit of upgrading the Central Registry version of HotSpot would be the numerous upgrades from the migration to the .NET Framework. The most current version of HotSpot, being a .NET Framework product, is fully supported in modern Windows systems unlike the Central Registry version. As described in Section 3.5, other modernization upgrades have been implemented beyond those required for the .NET Framework migration such as improved graphics rendering, better performance, and significantly improved exception (error) handling.

#### **5.6 Consistency between Central Registry Toolbox Version 2.07.1 and the Latest Versions of HotSpot**

HotSpot uses the well-documented Gaussian Equation<sup>11,12</sup> for all atmospheric dispersion calculations of radiological material from an accidental release and it configures that equation into four different release scenarios:

- Release from a standard geometric point, line, or area
- Release from a fire
- Release after an explosion

- Resuspension of material deposited on the ground

As described in Section 2, the only enhancements that affected the above listed scenarios came in the upgrade with version 2.07.2 to the explosion model. That upgrade only added new options and did not change the underlying physics of the explosive scenario. For that reason, running both the Central Registry and latest versions of HotSpot with the equivalent input parameters, will generate the same numerical results. Therefore, there is no numerical benefit to running the Central Registry version over the latest version of HotSpot.

## 5.7 SQA Procedures

All the releases of HotSpot, since version 2.07.1, have strictly followed the SQA procedures for testing. These procedures were established during the development of SQA documents for submission of version 2.07.1 into the Central Registry. One of these testing procedures is to run HotSpot's internal quality assurance verification routines. These routines run examples of the above listed fundamental HotSpot scenarios. Since the Central Registry version of HotSpot 2.07.1, these internal checks have remained unchanged except for the example explosion scenario, which was updated to use the new explosive options put in with version 2.07.2. Therefore, with the exception of the explosion example, the analytical answers to the internal verification routines are also consistent between the Central Registry and all subsequent versions.

The consistency in both the SQA verification and scenario physics between the latest version of HotSpot and the Central Registry version underscores two key points.

- The success of the modernization effort, despite extensive modification to the entire code base, in achieving a version compatible within the new .NET Framework environment while preserving both HotSpot's core mathematical calculations and output results to be consistent with the Central Registry version of HotSpot.
- There is no valid reason in terms of HotSpot itself, i.e., in terms of its calculation methodologies or its user interface, to use an older version of the program especially when there are known changes to the Microsoft Windows that restrict the use of older VB6 era applications such as HotSpot version 2.07.1.

## Section 6

---

### *Description of Accompanying SQA Documentation*

---

This section summarizes the available HotSpot SQA documentation that directly describes the HotSpot Support and Development team's adherence to the ten work activities required by DOE O 414.1D Attachment 4 (Requirement 4, a through j):

- HotSpot Software Configuration Management Plan (SCMP)
- HotSpot Software Test Plan (STP)

The SCMP addresses HotSpot's activities in Project Management and Quality Planning, Risk Management, Requirements Identification and Management, Software Configuration Management, Software Safety, and Problem Reporting and Corrective Action. The structure of the SCMP follows an IEEE standard template for software configuration management plans.<sup>13</sup>

The STP describes HotSpot procedures in the area of Verification and Validation. The structure of this document is based on an IEEE standard for software test documentation.<sup>14</sup>

The HotSpot User's Guide is applicable for addressing SQA work activity 10: Training of Personnel in the Design, Development, Use, and Evaluation of Safety Software. HotSpot's Help section, an inline HTML based set of documentation, available in the main HotSpot interface complements the information in the User's Guide. New training material, based on the latest versions of HotSpot, has been developed as well and been presented to users.

# Appendix A

## *HotSpot Release Notes*

---

### **HotSpot 2.07.2 – September 2011**

- Full Vista and Windows 7 compatibility. Default installation directory supports text file output.
- HotSpot Help display correction.
- New Automatic Batch Mode. HotSpot can be run in a “background” mode that produces Table output and Google Earth .KML files without displaying the HotSpot graphical user interface. When this mode is enabled, HotSpot can be launched at the command prompt or a batch file. These files are placed in the HotSpot “Output” folder.
- New Explosive Release ballistic particle option. This option improves the modeling of particles with aerodynamic diameters greater than 100 micrometers. This new option is turned off by default.
- Supports MACCS2 meteorological file input in both wind-from and wind-towards format when site-specific meteorological data are input into HotSpot for calculation of percentile dose values.
- New resuspension factors developed by Maxwell and Anspaugh (2010).
- New Explosive Release cloud rise calculation. This option is turned off by default. New cloud rise formulas are most appropriate for detonations of about 100 pounds or less. Therefore, the old Church formula is still used above 100 pounds, even when the new option is selected.

### **HotSpot 3.0.0 – March 2013**

#### New Features:

- Complete rewrite of code to make it compatible with Microsoft's .NET Framework. From a user's perspective only, minor interface changes result from the rewrite.
- New capability: Surface Roughness entry in the Setup/Terrain section. User can enter a surface roughness value to modify the calculated concentrations. Only applicable when Standard terrain is selected.
- Accessibility of Explosive Release – Ballistic Particle option. This option (implemented in version 2.07.2) is now available through the user interface. Ballistic interface is available with explosive release having airborne fractions less than one (the source has a non-respirable component) through the Setup – "Change/View AF Distribution" option.
- HotSpot Output Directory was modified to reflect version 3.0. Output directory will now appear as “HotSpot30” under a parent folder as given by the Windows Environment variable “AppData”. Note that all sub-directories and files under HotSpot30 are 100% compatible with HotSpot2.07 versions. Files from 2.07 versions need only be moved to their corresponding location under the new HotSpot30 folder for use in this new version.



- Digital Signature was applied to the installer and program assuring the Windows environment that HotSpot comes from a trusted source. In Windows 7, not having a digital signature causes a warning box to come up whenever HotSpot is run.
- New Explosive Release cloud rise calculation option, implemented in version 2.07.2, set to be the default option.

#### Bug Fixes:

- Several bug fixes. The most notable being a fix to a user change of the DCF library when a user does not follow through with a selection of a new source. Also, when a user selects a nuclide, they can let HotSpot automatically select the conservative default solubility classification recommended by DOE for a particular nuclide. Another notable fix was done on the international mode, which would only affect users outside the U.S.

### HotSpot 3.0.1 – September 2013

#### Bug fixes:

- Fixed three significant Windows 7 issues:
  1. Executable was installed properly under "Program Files/HotSpot 3.0." Installation package now smaller: removed dependency on older Windows DLL files.
  2. Forms are now properly scaled based on DPI instead of font for users who change their screen resolution to something other than 100%. This is a more common feature in Windows 7.
  3. Windows installer was starting every time most users started the program, even after HotSpot was already installed. Now, no user should see the installer come up more than once over many uses.
- GPS capability was updated to directly use a serial port through a USB connection. Non-USB connections should still work.
- Fixed table output inconsistencies for fire scenarios.

### HotSpot 3.0.2 – July 2014

#### Bug fixes:

- Ground Shine Dose rate factor for ICRP 30 with international (non-English) Windows environment\*

In international mode, a specific ICRP 30 ground shine dose conversion factor was being initialized incorrectly, which resulted in incorrect doses for any scenarios involving ground shine from the ICRP 30 library. Two of the Quality Control tests, the General Explosion and the General Fire scenarios, were therefore failing the Quality Control test.

- **KML Writer for Google Earth Output\***

In international or non-USA mode, the format for contour coordinates in Google Earth KML files was based on the Windows Language settings and Google Earth was not able to display the data. Contour coordinates in KML files are now always formatted with English settings, which seems to be what Google Earth always expects. Contour information such as the legend table is still based on the Windows Language settings.
- **Distortion of small images in the Map window (either width less than 793 pixels and/or height less than 600 pixels)**

When importing such background images into the Map Window, they were being stretched and distorted to fill up the entire window. This caused latitude and longitude coordinates to be calculated incorrectly.
- **Display of Contour legend information for two levels in Map Window**

In scenarios where the highest of three contours was not reached, leaving only two valid contours to plot, the legend information for the two remaining contours was displayed in reverse colors, although the map or plotted contours were being displayed correctly.
- **Print to clipboard option, which was not working in Contour and Centerline display windows**
- **Ability to generate textual output of the contours (PLM files) in latitude/longitude coordinates as opposed to x/y**
- **User defined wind direction , which was not being saved between sessions in non-USA mode\***
- **Inconsistencies for users canceling out of inputting wind speed and direction values**

\* This bug would not affect users of HotSpot within the U.S.

### HotSpot 3.0.3 – August 2015

#### Bug fixes:

- The GPS import algorithm was not capturing all the valid NEMA data being streamed through a USB device. This same routine was also delaying data processing by waiting for the input buffer to completely fill. This wait was reduced hence speeding up GPS data processing by a factor of 3 to 4.
- Import of mixture files written in international (non-English) format was not working.
- Acute Mixture Files - Header was not being written correctly: The *Mixture Scaling Factor* was not being written and an extra blank line was being added. Also eliminated a function which was adding the mixture scaling factor to pre-existing mixture files not containing that field.
- Algorithm for re-calculating contours levels was failing because it was based on behavior of the old Visual Basic 6 IDE.
- Eliminated the display of the *About* form (a splash Screen) in automated or non-interactive mode.
- When using the compass option for table output, upwind deposition values for the Explosive Release Model was incorrectly outputting non-zero concentrations

- The compass option for the display of contours from a nuclear explosion was not in effect.
- The Quality Control test was not consistent for nuclear explosions as it was using a variable based on the most recent scenario as opposed to a constant value.
- Receptor locations on contour plots were not being displayed correctly.
- Table Output for Nuclear Explosion Model was not re-setting centerline distances correctly when English distance units were set.

Eliminated the following Windows XP dependencies:

- Migrated from Visual Studio (VS) 2008 to VS 2012 and from development in a Windows XP environment to Windows 7.
- Removed reliance on installer software integrated into Microsoft's Visual Studio IDE. Developed a new deployment process using third-party installer software program.

### HotSpot 3.1.0 – October 2017

New Features:

- 95<sup>th</sup> Percentile Upgrade
  - Upgraded 95<sup>th</sup> percentile capability to include DOE-STD-3009 and NRC Guide 1.14510 Position 3.
  - New interface for entering distances that vary with each compass sector. Includes options to export and import distances from a file.
  - Added several new distance units.
  - Added an option to calculate and output the maximum doses in each compass sector.
  - Added a new progress bar during the percentile calculations.
- Contour Graphics Options
  - Reworked all contouring graphics (plume centerline, compass based, map or imagery based, and google formatted) to allow modernized fill and transparency routines.
  - Replaced all color selection interfaces with a modern Windows color selection dialog.
- Acute Dose Capability
  - Implemented the Oak Ridge National Laboratory 1-day and 4-day Acute Inhalation Dose Conversion Factors (DCFs) Library (Eckerman, 2012; EPA 2002).
  - Updated lethality levels for critical organs.
  - Updated the acute dose example scenario.
  - Replaced the acute options form with a new window adding more flexibility in selecting RBE (Relative Biological Effectiveness) settings.
  - Added a new context sensitive help form for explaining RBE settings.
  - Added 1-day dose values for Cf-252 which were not in the original database.
  - Added better consistency to the display of a user's selected acute dose options.

## Improvements

- Modernized the centerline graphics algorithms and interface. Updated the menu options and added several application settings for saving a user's preferences.
- Modernize Map Form File menu options and algorithms. Also added the ability to change background color of the text section.
- Modernized the GPS Setup interface.
- Replaced the use of a file for storing GPS options with application settings.
- Streamlined deposition button changes which occur when a scenario toggles deposition on or off.
- Removed various obsolete variables and references.
- Separated the textual (PLM) and Google output interfaces. These two formats had been accessible through one button which toggled back and forth between them.
- Added the output Air temperature value in table output for fire scenarios.

## Bug Fixes

- Removed Centerline Graphics from the QC test. This eliminated a conflict between the new graphics routines and the full QC test.
- Three issues with the center line plots "Show All Stability Classes" option:
  - The maximum dose value was not being re-calculated under certain conditions.
  - The displayed textual value for the adjusted wind speed was not correct.
  - The six stability curves were not being labeled.
- Fixed the Rain out Coefficient, which was not processing user input properly for certain values.
- Corrected a navigator error was occurring during nuclear explosion prompt effects calculations.
- Acute Dose Critical Lung statement in table output was not being output when English units was selected.
- Fixed an incorrect acute lethality level for the small intestines when the IAEA 30-day doses were being used.
- Corrected managing imagery that had not been geo-registered was leading to an erroneous state
- Fixed the "Cloud Top" parameter that was being displayed in table output for fire scenarios not based on the cloud top option.
- Small, non-zero concentrations were being generated slightly upwind of an explosive release.
- Fixed the Nuclear Explosion - Compass contour plots button was not behaving properly.

## HotSpot 3.1.1 – July 2019

### Bug fixes:

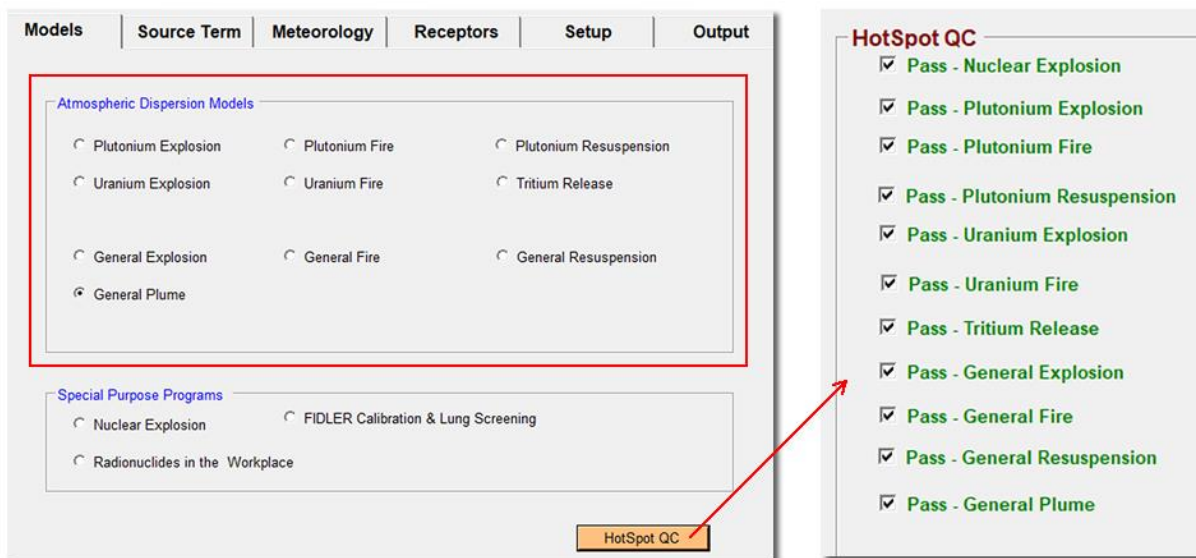
- Corrected UI settings for the airborne and release fractions during the creation of a mixture file

- Restored the ability to run up to twenty wind sector-independent distances in the Percentile Meteorological UI
- Corrected the processing of the last nuclide in a mixture file containing 50 nuclides
- Fixed to two issues related to non-USA language settings:
  - Release coordinates written to Google KML files now correctly in English format, as required by KML, instead of in the language settings of the Window OS
  - Mishandling of numbers with language settings that do not use a period as a numerical symbol, e.g., decimal or thousands separator
- Changed the wind groupings display in the Percentile Meteorology UI to correctly reflect the speed ranges used in the code
- Fixed the display in the Mapping UI of an imported NARAC plume file
- Eliminated an error in the centerline graph UI from selecting the “Show Scenario” option with an airborne fraction less than or equal to 0.01
- Safeguarded against a possible error in an initialization file when processing an empty string
- Removed an obsolete text field in the Fidler UI relating to lung screening
- Removed a nuclide which contains no doses in DCFPAK 3.0 (Nb-97m)
- Certain GUI paths not setting the enabling/disabling of deposition buttons
- Certain GUI paths not displaying the correct value of the sigma theta parameter (G stability)
- Inhalation dose conversion factors missing for two special chemistry nuclides
- Output from 95th Percentile runs was not providing enough scenario information to properly QA scenarios
- Closing of the main window was triggering some unnecessary code processing
- The ordinate axis of the centerline graphic had an incorrect deposition label under SI units

## Appendix B

### *Equivalence Verification between the Latest Version of HotSpot (3.1.1) and the Central Registry Version (2.07.1)*

#### Background



**Figure B.1.** HotSpot's *Models* user interface for selecting an atmospheric dispersion model. To the right, the *HotSpot QC* button runs a series of validation tests covering each model option, based on the example scenario set used in this section.

HotSpot's primary usage as well as most of the application's scientific methods focus on the various Atmospheric Dispersion Models<sup>1</sup> shown in Figure B.1. User customization of any of these dispersion models is processed as a *scenario*, with the parameters or data for a scenario coming from either the user interface (UI) or a stored text file, known as a *scenario file*. Besides user-generated scenarios, HotSpot provides a set of example scenarios, in the text-file format, with one example for each of the dispersion models as well as two specialized scenarios addressing the acute dose and 95<sup>th</sup> Percentile modules. This extensive coverage makes the example scenario set a valuable SQA tool; it is used internally within both HotSpot's verification and validation test suites, one of which is available to users through the *HotSpot QC* button, see Figure B.1. This SQA framework was in place prior to the Central Registry version of HotSpot (2.07.1) and remains unchanged in the most current HotSpot version (3.1.1).

This section will use a similar approach to verify the equivalence between the latest HotSpot version (3.1.1) and Central Registry version of HotSpot (2.07.1), by providing step by step procedures for using the example scenario set as a benchmark.

<sup>1</sup> - A characterization of an atmospheric dispersion hazard based on a source material and release method such as a point, line, area (Term or Continuous Release Models); fire; spill; or explosive release mechanism.

There are twelve example scenarios, all listed in Table B.1 along with corresponding instruction sections. The basic steps outlined in General Procedures (below) are adequate to show equivalence between the two versions for eight of the twelve scenarios. The remaining four scenarios follow these same general procedures but require additional instructions for setting the environment within the latest version to match that of the Central Registry version. The following instructions titled Explosion Scenario Procedure, General Explosion Comparison (I-131), and Acute Scenario Comparison provide explanations for those initial differences.

**Table B.1. Twelve Scenarios, Instruction Sections, and Checklist**

Instruction Section	Scenario <small>(C:\Program Files (x86)\Hotspot 2.07.1\HotSpot20\Scenario\Examples)</small>	Match <small>(Between v2.07.1 &amp; v3.1.1)</small>
General Procedures	General Fire	<input type="checkbox"/>
	General Plume	<input type="checkbox"/>
	General Resuspension	<input type="checkbox"/>
	MACCS2 Pu-239 M Comparison	<input type="checkbox"/>
	Plutonium Fire	<input type="checkbox"/>
	Plutonium Resuspension	<input type="checkbox"/>
	Tritium Release	<input type="checkbox"/>
	Uranium Fire	<input type="checkbox"/>
Explosion Scenario Procedures	Plutonium Explosion	<input type="checkbox"/>
	Uranium Explosion	<input type="checkbox"/>
General Explosion Comparison (I-131)	General Explosion	<input type="checkbox"/>
Acute Scenario Comparison	Acute Pu-238 100% Lethality at 0.50 km	<input type="checkbox"/>

## General Procedures

1. If necessary, install both the Central Registry and latest version of HotSpot (versions 2.07.1 and 3.1.1) on the same computer. Both are available at:

<https://naracweb.llnl.gov/web/hotspot/registerUser.html>

2. Open both versions simultaneously, making sure to open 2.07.1 with elevated privileges<sup>2</sup> (*right mouse click → Run as Administrator*).

3. Use version 2.07.1 to open the first example scenario in Table B.1.

a. Use the menu option *File → Open Scenario* and browse to the Examples folder

b. Select the first scenario listed in Table B.1 (*General Fire*).

4. Use version 3.1.1 to open the same example scenario in Table B.1.

a. Use the *File → Open Scenario* option as well but by default, you will start in the Scenario directory of this later version.

b. Browse to the example scenario folder of version 2.07.1 (Or copy the path below into the “*File Name*” field in the dialog) :

[C:\Program Files \(x86\)\Hotspot 2.07.1\HotSpot20\Scenario\Examples](C:\Program Files (x86)\Hotspot 2.07.1\HotSpot20\Scenario\Examples)

c. Open the same General Fire scenario from Step 3.

Depending on your system settings, you may or may not be able to browse to the above directory due to its restricted access (See footnote 2 or Section 5.4). If this is the case, restart HotSpot version 3.1.1 with elevated privileges using the same procedure for opening 2.07.1 (*right mouse click → Run as Administrator*).

5. Generate output for both version 2.07.1 and 3.1.1.

a. In each version, click on the *Output* tab.

b. Click on the *Table Output* button

6. For the selected scenario, verify that both versions generate the same output

a. Inspect the following information visually or use an application that shows differences between a pair of files:

All numerical data in the header:

- Parameter values

---

<sup>2</sup> Necessary to avoid the restriction with 2.07.1 discussed in Section 5.4.



- Maximum dose level and downwind distance to that value
- Contour extents

All values listed in the twenty-receptor table:

- TEDE doses in rem
- Respirable time-integrated air concentrations in (Ci-sec)/m<sup>3</sup>
- Deposition values, if they exist, in uCi/m<sup>2</sup>
- Ground Shine Dose Rates, if they exist, in rem/hr
- Arrival Times

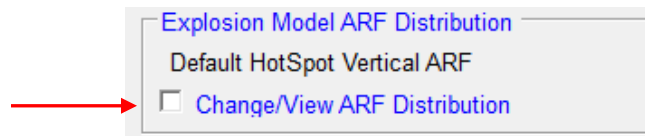
Note - Some of the numerical formats have changed. We are verifying numerical values only and not formats. (e.g. 1.0 = 1.00E+00)

7. Check the box in the “Match” column for the General Fire scenario in Table B.1.
8. Repeat steps 3 through 6 for the remaining scenario files in Table B.1:
  - General Plume
  - General Resuspension
  - MACCS2 Pu-239 M Comparison
  - Plutonium Fire
  - Plutonium Resuspension
  - Tritium Release
  - Uranium Fire

## Explosion Scenario Procedures

During two of the intervening releases between 2.07.1 and 3.1.1, a new option for calculating the height of an explosive cloud, the Green Field option, was added (see version 2.07.2 release notes) and eventually set as the default method (see version 3.0.0 release notes). However, the Church formula is still available in version 3.1.1 and the new Green Field default can be easily turned off to match the environment of the Central Registry version.

1. Follow steps 3 and 4 from General Procedures to set up the comparison for the first scenario under Explosion Scenario Procedures in Table B.1, *Plutonium Explosion*.
2. Prior to generating output for 3.1.1, turn off the newer Green Field option:
  - a. Open the Explosive Model Options (*Setup* Tab) by checking the *Change/View ARF Distribution* box shown below:



- b. In the Cloud Distribution Window, uncheck the Green Field Option:



- c. Click the *Return* button to return to the main interface.
3. Generate output for both version 2.07.1 and 3.1.1.
4. Verify the consistency between both versions for the *Plutonium Explosion* scenario and check the “Match” column in Table B.1.
5. Repeat Steps 1-4 for the next scenario, *Uranium Explosion*.

## General Explosion Comparison (I-131)

There is an added discrepancy with this scenario due to the half-life of the source nuclide, I-131. Version 2.07.1 does not time decay nuclides with half-lives over a certain threshold, which was arbitrarily chosen to be 7 days approximately. The half-life of I-131 is slightly over 8 days, just above the 7-day threshold. This limit existed in 2.07.1 only to save computational resources and was removed in a subsequent version. Version 3.1.1 therefore time-decays all nuclides regardless of half-life causing the slight deviation with 2.07.1 when using I-131 as the source nuclide. The difference is only noticeable at large downwind distances with a plume arrival time close to the I-131 half-life.

1. Open the *General Explosion* scenario listed in Table B.1 in both version 2.07.1 and 3.1.1.
2. In version 3.1.1, follow the steps in Explosion Scenario Procedures to turn off the *Green Field* option.
6. Change the source nuclide in both versions to use a nuclide other than I-131.
  - a. Under the source term tab, change the source term nuclide to one that is either:
    - i. Below the 7-day half-life threshold (e.g., I-124 [half-life of 4.18 days], Cs-132 [6.475 days], etc.), or
    - ii. Well above the 7-day threshold (e.g., Cs-137 [half-life of 30 years], Co-60 [5.3 years], etc.)

In other words, only nuclides with half-lives close to and slightly above will see a small discrepancy.

- b. When re-selecting your nuclide in version 2.07.1, you will need to reset the source amount, in this case 30 Curies. This step is not necessary in version 3.1.1 because it no longer resets the source amount after changing a source term nuclide.
4. Generate output for both versions and record the results in Table B.1.

## Acute Scenario Comparison

This last comparison has one minor precaution due to its use of acute dose conversion factors. HotSpot does not provide for all the available acute dose parameters in the scenario file format. Those parameters are instead based on UI settings. Therefore, any comparison of acute dose scenarios must use the same UI inputs.

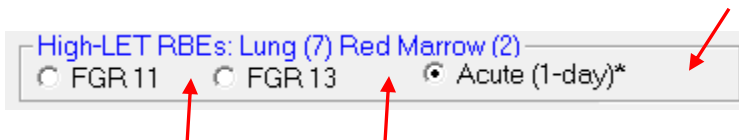
Also, version 2.07.1 offered two different 1-day databases for acute dose data with one database being removed prior to version 3.1.1. These steps will ensure the now obsolete database is not used in the comparison.

1. In both versions, open the acute dose scenario (*Acute Pu-238 100% Lethality at 0.50 km*) listed in Table B.1.
2. Ensure the obsolete, acute 1-day database is not selected in 2.07.1.
  - a. In 2.07.1, go to the Setup tab and find the dose database frame (see example below):



If the selected acute database is NOT “(1-day)\*” as shown (the asterisk distinguishes this database from the other 1-day option), then move on to step 3). If the 1-day option with an asterisk is selected, continue with part b)

- b. Click on any background area between the three labeled options (red arrows):



In the top section of the acute options interface, select any of the other four integration (dose database) options.

Select a dose integration time period, and click on the "Return" button.

Acute Intake - Dose in first day  
 Acute Intake - Dose in first 4 days  
 Acute Intake - Dose in first 30 days (Federal Radiological Assessment Center [FRMAC])  
 Acute Intake - Dose in first 30 days (IAEA-TECDOC-1432 Jan 2005)  
 Acute Intake - Dose in first day (DOE Vulnerability Assessment)

High Linear Energy Transfer (LET) Relative Biological Effectiveness (RBE)

**RBE - lung**

RBE = 1  
 RBE = 7 FRMAC, DOE V&A, NUREG/CR-4214, Rev. 2, Part I, 1993), and IAEA-TECDOC-1432  
 RBE = 10  
 RBE = 20

High Linear Energy Transfer (LET) Relative Biological Effectiveness (RBE)

**RBE - red marrow**

RBE = 1  
 RBE = 2 FRMAC, NUREG/CR-4214, Rev. 2, Part I, 1993), and IAEA-TECDOC-1432 (Jan 2005)  
 RBE = 3

Cancel Return

3. In version 3.1.1, ensure matching acute dose options with version 2.07.1

DCF Library

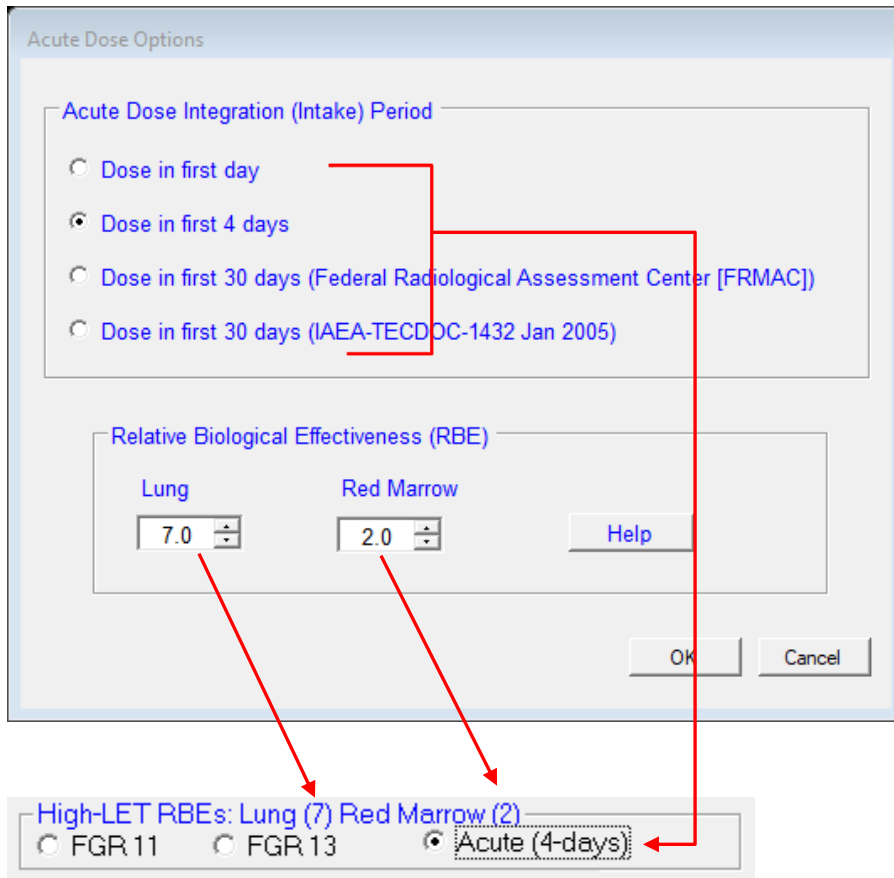
FGR 11  
 FGR 13  
 Acute (1-day)

options

a. Under the *Setup* tab, *DCF Library*, open the acute *options* form:

Based on the example below, set all three dose options in 3.1.1 to match those in 2.07.1:

- i. The database (integration period)
- ii. The lung RBE
- iii. The Red Marrow RBE



Version 3.1.1

Version 2.07.1

Note we are using the RBE settings in 2.07.1 from the title of the acute dose frame.

4. Generate output for both version 2.07.1 and 3.1.1 and record the results in Table B.1.

# Appendix C

## *Key Terms and Glossary*

---

ARG	Accident Response Group
CDC	Center for Disease Control
Deployment	The process of providing a working version of an application through an installer program.
Deprecated	Describes a computer language command or statement that will be made invalid or obsolete in future versions of the language as newer and more robust constructs have been provided. The period over which language statements remain available yet deprecated is intended to provide an opportunity for their replacement with the newer standards.
DHS	Department of Homeland Security
Digital Signature	A digital code or signature that can be attached to an electronically transmitted file, such as the HotSpot installer, and validates the identity of the sender.
DLL	Dynamic Link Library – A package of library functions used by Windows compiled programs during execution. Since these libraries are dynamic or called during runtime, they must be available in a user’s operating environment.
DOE	U.S. Department of Energy
DOE G	U.S. Department of Energy Guide
DOE O	U.S. Department of Energy Order
Event	An action associated with a GUI or component therein, such as a mouse click or closing of a window. A user normally drives an event either directly or indirectly.
Event-driven Programming	The concept of application code sections that are activated by an Event.
Event Handling	See Event-driven Programming
Exception	An error during the execution of a program or application
EPA	Environmental Protection Agency
FBI	Federal Bureau of Investigation
FEMA	Federal Emergency Management Agency
File Dialog Interface	Provides an interface to a user for the purpose of opening or saving a file on a computer or storage device.

Form	The main window or interface through which a user interacts with an application. Can contain a GUI or other visual display screen. Also called a window, which should be distinguished from Windows, the Microsoft Operating System.
GIS	Graphical Information System
GPS	Global Positioning System.
Graphics	Refers to the drawing of an object on a form or window such as a line, arc, ellipse, rectangle, as well as text. Graphics objects are used in HotSpot to build displays of concentration and other output.
Ground Shine Dose	Dose due to radioactive materials deposited on the ground.
GUI	Graphical User Interface
ICRP	International Commission on Radiological Protection
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
KML files	Keyhole Markup Language. An XML based file format used in Google Earth and other location rendering applications.
LLNL	Lawrence Livermore National Laboratory
Load	The event associated with the opening of a form
Module	Used in both the VB6 and VB.NET languages, describes a file or entity used only for the declaration and initial setting of variables. No logical code constructs beyond that are allowed within a Visual Basic module. Modules are outdated, as they do not follow modern Object-Oriented programming conventions; variables declared within a module can have global scope resulting in their accessibility among multiple objects.
NARAC	National Atmospheric Release Advisory Center: Provides tools and services that predict and map the spread of hazardous material released into the atmosphere. NARAC is a national support and resource center for planning, preparedness, real-time emergency response, and detailed assessments of threats and/or incidents involving a wide variety of hazards for over 30 years. For more information see <a href="https://narak.llnl.gov/">https://narak.llnl.gov/</a>
.NET Framework	An infrastructure and environment within the Microsoft Windows Operating System. One of the .NET Framework's primary function is to facilitate the development, deployment, and running of applications.
NTNF	National Technical Nuclear Forensics (Center)
OS	Operating System
PowerPacks	A Microsoft Library or DLL providing graphical and printing capabilities.
RAP	DOE/NNSA's Radiological Assistance Program:



<https://nnsa.energy.gov/aboutus/ourprograms/emergencyoperationscounterterrorism/respondingtoemergencies/firstresponders-0>

Safety Basis	The documented safety analysis and hazard controls that provide reasonable assurance that a DOE nuclear facility can be operated safely in a manner that adequately protects workers, the public, and the environment. ( <a href="http://definitions.uslegal.com/s/safety-basis/">http://definitions.uslegal.com/s/safety-basis/</a> )
VB6	Visual Basic 6. The last version of Microsoft Windows Visual Basic pre-dating the .NET Framework programming environment.
VB.NET	Visual Basic.Net; The Microsoft Windows .NET Framework version of Visual Basic
Windows	Any of the Microsoft Windows Operating Systems
Window	See form
Windows7	A commonly used version, ongoing within the DOE community, of the Microsoft Windows Operating System since 2010.
Windows XP	A commonly used version, within the DOE community, of the Microsoft Windows Operating System from 2001 to 2014.
Wrapper	In general, software or code designed to provide a convenience, compatibility, or similar benefit. As used in this document, describes VB.NET code designed to preserve or simulate the behavior of a VB6 code block. The use of wrappers in HotSpot 3.0 (VB.NET) and later versions provides a consistent way to achieve the results originally intended in the VB6 version and at the same time avoid deprecated VB6 constructs.

# References

1. U.S. Department of Energy, "DOE Standard Preparation of Nonreactor Nuclear Facility Documented Safety Analysis," DOE-STD-3009 -2014, Washington, 2014.
2. *HotSpot Software Configuration Management Plan*, LLNL-SM-641990, LLNL, July 22, 2013.
3. *HotSpot Software Test Plan*, LLNL-SM-641989, LLNL, July 22, 2013.
4. DOE O 414.1D, Quality Assurance, April 25, 2011.
5. *DOE G 414.1-4, Safety Software Guide for Use with 10 CFR 830 Subpart A, Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance*, DOE G 414.1-4, U.S. DOE, 2005.
6. Homann, S.G., *HotSpot Health Physics Codes User's Guide*, Lawrence Livermore National Laboratory, Report LLNL-SM-636474, 2013.
7. Sharon, Avi, Itzhak Halevy, Daniel Sattinger, and Ilan Yaar. "Cloud rise model for radiological dispersal devices events." *Atmospheric Environment* 54, 603-610, 2012.
8. *Support Statement for Visual Basic 6.0 on Windows Vista, Windows Server 2008, Windows 7, Windows 8 and Windows 8.1, Windows Server 2012, and Windows 10*, August 2015, <https://msdn.microsoft.com/en-us/vstudio/ms788708>
9. *Description of the Cumulative Update Rollup for the Visual Basic 6.0 Service Pack 6 Runtime Extended Files*, 2014. <https://support.microsoft.com/en-us/kb/957924>
10. *U.S. Nuclear Regulatory Commission (NRC) Regulatory Guide 1.145*, U.S. NRC, Rockville: NUS Corporation, 1981.
11. Gifford, F. A., Jr., "Atmospheric Dispersion Models for Environmental Pollution Applications," *Lectures on Air Pollution and Environment Impact Analyses*, p. 3558, American Meteorological Society, Boston, MA, 1975.
12. Pasquill, F., *Atmospheric Diffusion*, 2nd. ed., John Wiley & Sons, New York, 1974.
13. *IEEE Std 828-2005, Standard for Software Configuration Management Plans*, IEEE, 2005, IEEE Standard 828-2005, IEEE, 2005.
14. *IEEE Std 829-1998, Standard for Software Test Documentation*, IEEE, 1998, IEEE 829-2998, IEEE.
15. *Visual Basic*, April 18, 2016, [https://en.wikipedia.org/wiki/Visual\\_Basic](https://en.wikipedia.org/wiki/Visual_Basic).

16. *Visual Basic .NET*, April 18, 2016, [https://en.wikipedia.org/wiki/Visual\\_Basic\\_.NET](https://en.wikipedia.org/wiki/Visual_Basic_.NET).

17. Wikipedia, *Comparison of Visual Basic and Visual Basic .NET*, February 13, 2016, [https://en.wikipedia.org/wiki/Comparison\\_of\\_Visual\\_Basic\\_and\\_Visual\\_Basic\\_.NET](https://en.wikipedia.org/wiki/Comparison_of_Visual_Basic_and_Visual_Basic_.NET)