



**Csv_to_ZDIFF: A Translator in the
CASRED Tool Set, Version 1.4**

by Geoffrey C. Sauerborn

ARL-TN-0261

September 2006

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

DESTRUCTION NOTICE—For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5069

ARL-TN-0261

September 2006

Csv_to_ZDIFF: A Translator in the CASRED Tool Set, Version 1.4

Geoffrey C. Sauerborn
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.				
1. REPORT DATE (DD-MM-YYYY) September 2006		2. REPORT TYPE Final		3. DATES COVERED (From - To) May to July 2006
4. TITLE AND SUBTITLE Csv_to_ZDIFF: A Translator in the CASRED Tool Set, Version 1.4			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Geoffrey C. Sauerborn (ARL)			5d. PROJECT NUMBER AH80	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory Weapons and Materials Research Directorate Aberdeen Proving Ground, MD 21005-5069			8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TN-0261	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT This document describes a translator that converts a comma-separated value format file into a "Z-data" ASCII (American standard code for information exchange) format. Called Csv_to_Zdata, the translator is software tailored to work within the Casualty Reduction (CASRED) model tool set. CASRED is a U.S. Army item-level lethality and survivability performance model that calculates the probability of personnel incapacitation, given a hit. This document describes Csv_to_Zdata, its options, and input but does not address CASRED.				
15. SUBJECT TERMS CASRED; spreadsheet; translator; ZDIFF				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 20
a. REPORT UNCLASSIFIED	b. ABSTRACT UNCLASSIFIED	c. THIS PAGE UNCLASSIFIED		
			19b. TELEPHONE NUMBER (Include area code) 410-278-9076	

Contents

List of Figures	v
List of Tables	v
Acknowledgments	vi
1. Introduction	1
2. The Excel Fragmentation File	2
2.1 Excel File Overview.....	3
2.2 Angular Segments in the Excel File.....	3
2.3 The Shaping Factor	4
3. The CSV and Excel Fragmentation File Schema	5
4. Csv_to_Zdata the Translator	5
4.1 Compiling Csv_to_Zdata	6
4.2 Running Csv_to_Zdata Interactively.....	7
4.3 Running Csv_to_Zdata in Batch Mode	7
5. Translation Rules	7
6. Csv_to_Zdata Configuration File and Parameters	9
6.1 Munition_Name.....	9
6.2 Classification_Name	10
6.3 Output_Format_Name.....	10
6.4 Output_Format_Numeric_Fixed	10
6.5 Shaping_Factor_Convert.....	10
6.6 Conversion_Factor_Angles.....	10
6.7 Conversion_Factor_Velocity.....	10
6.8 Conversion_Factor_Mass.....	10
7. Summary	11

8. References	12
Appendix A. Comma-Separated Values (CSV) Library	13
Distribution List	15

List of Figures

Figure 1. Where Csv_to_Zdata fits in the translation process.	1
Figure 2. Blast fragmentation profile describes blast fragments.	2
Figure 3. Angular segments from blast center.	2
Figure 4. Angular segments in the spreadsheet.	3
Figure 5. A highlighted angular segment record header.	4
Figure 6. Fragmentation file format description.	5
Figure 7. Fragmentation file in .csv format.	6
Figure 8. Csv_to_Zdata software directory organization.	6
Figure 9. A Z-data file after a successful translation (a fixed length ASCII record format).	8
Figure 10. Csv_to_Zdata configuration file.	9

List of Tables

Table 1. Default units of measure.	9
---	---

Acknowledgments

The author would like to acknowledge the contributions of Richard Pearson of the U.S. Army Research Laboratory (ARL), who not only completed a thorough technical review but provided expert consultation and research regarding points in the Joint Munitions Effectiveness Manual.

The author would also like to thank Nancy Nicholas of ARL for her superior editorial review and editing to which many have grown accustomed.

1. Introduction

Csv_to_Zdata is a translator that converts a comma-separated value format file into a “Z-data” ASCII (American standard code for information exchange) format. Z-data files, which are used to share a warhead’s fragmentation patterns, are used by a number of U.S. Army analytical tools directly or in conjunction with a separate pre-processing tool.

The Csv_to_Zdata tool is designed to translate fragmentation pattern information for use in the Casualty Reduction (CASRED) model. CASRED is a U.S. Army item-level lethality and survivability performance model that calculates the probability of personnel incapacitation (*I*). The U.S. Army Research Laboratory has created a number of pre- and post-processing software tools to supplement CASRED and allow it to be interfaced with other simulations¹. One of the pre-processing tools is “Zdata_to_CASRED” which translates normal Z-data files into fragmentation pattern files that can be directly entered into CASRED. The output of Csv_to_Zdata will be the input for Zdata_to_CASRED which in turn will produce a file that will be used directly by CASRED. This document describes Csv_to_Zdata, its options, and input but does not address CASRED.

Figure 1 portrays the translation sequence. The fragmentation profile is stored as a spreadsheet in the Excel² format. This spreadsheet and its fields are described later in detail. The spreadsheet must first be exported into the comma-separated value (csv) format (2). This can be accomplished with the “save as” feature common to many spreadsheet applications. In the figure 1 final sequence, Csv_to_Zdata translates the .csv format to the Z-data fixed field length format.

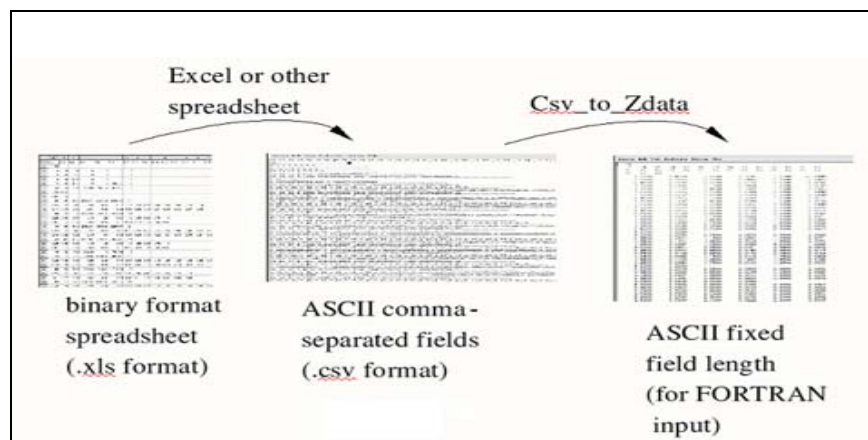


Figure 1. Where Csv_to_Zdata fits in the translation process.

¹The particular CASRED tool that uses the Z-data file is called Zdata_to_CASRED_Metric.

²Excel is a trademark of Microsoft.

2. The Excel Fragmentation File

This section provides a detailed explanation of the fragmentation file and its logical fields. The fragmentation profile describes fragment mass and velocity distributions following a munition detonation. Targets may be placed about the point of detonation to determine probability of damage as in figure 2.



Figure 2. Blast fragmentation profile describes blast fragments.

Conceptually, the fragments are described in angular segments about the point on the ground directly below the detonation point, known as ground zero (GZ). The angular segments are measured about the GZ in the ground plane from the long axis of the warhead. Angles are measured from zero degrees at the front of the warhead around to 180 degrees at the rear. The angular segments pictured in figure 3 are shown in five 5-degree intervals. The actual segment intervals could be any angle and do not have to be equally spaced. The fragment data only span 180 degrees since the remaining fragment profile (as high as 360 degrees) is assumed to be symmetric.

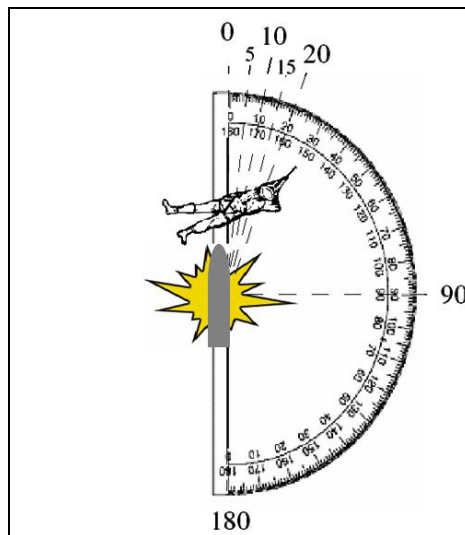


Figure 3. Angular segments from blast center.

2.1 Excel File Overview

Viewed with a spreadsheet application, the .xls fragmentation file might appear similar to the file seen in figure 4. The first row contains text headers for each column (e.g., “X.1”, “X.2”, and so on for every column in the file that contains data). This header row has no meaning to the context of the data but must be present. The number of angular segments (in this case, 36) is identified in row 2 of column A.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	X.1	X.2	X.3	X.4	X.5	X.6	X.7	X.8	X.9	X.10	X.11	X.12	X.13	X.14	X.15
2	36														
3	0	5	2.5	0	0	0	0								
4	5	10	7.5	0	0	0	0								
5	10	15	12.5	0	0	0	0								
6	15	20	17.5	0	0	1293.55	0								
7	20	25	22.5	1293.55	2587.11	2570.71	2								
8	1.55	1.65													
9	0.03	0.04													
10	25	30	27.5	2570.71	2554.32	2558.59	14								
11	0.25	0.35	0.45	0.55	0.65	1.05	1.15	1.25	1.35	1.45	1.55	1.65	1.75	3	
12	0.13	0.09	0.06	0.04	0.03	0.04	0.06	0.09	0.14	0.21	0.36	0.48	0.07	0.06	
13	30	35	32.5	2558.59	2562.87	2530.84	11								
14	0.25	0.35	0.45	1.15	1.25	1.35	1.45	1.55	1.65	1.75	3				
15	0.07	0.05	0.03	0.03	0.05	0.07	0.11	0.19	0.25	0.04	0.03				
16	35	40	37.5	2530.84	2498.81	2297.87	24								
17	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95	1.05	1.15	1.25	1.35	1.45	1.55	1.6
18	0.46	0.32	0.22	0.15	0.1	0.08	0.07	0.09	0.13	0.2	0.3	0.47	0.75	1.24	1.6
19	40	45	42.5	2297.87	2096.93	1048.46	11								
20	0.25	0.35	0.45	1.15	1.25	1.35	1.45	1.55	1.65	1.75	3				
21	0.07	0.05	0.03	0.03	0.05	0.07	0.12	0.2	0.26	0.04	0.03				
22	45	50	47.5	1048.46	0	754.34	0								
23	50	55	52.5	754.34	1508.67	1805.48	18								
24	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95	1.05	1.15	1.25	1.35	1.45	1.55	1.6
25	0.19	0.13	0.09	0.06	0.04	0.03	0.03	0.04	0.05	0.08	0.12	0.19	0.3	0.5	0.6
26	55	60	57.5	1805.48	2102.3	1716.94	13								
27	0.25	0.35	0.45	0.55	1.05	1.15	1.25	1.35	1.45	1.55	1.65	1.75	3		
28	0.1	0.07	0.05	0.03	0.03	0.04	0.07	0.11	0.17	0.28	0.37	0.05	0.05		
29	60	65	62.5	1716.94	1331.58	1428.62	24								
30	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95	1.05	1.15	1.25	1.35	1.45	1.55	1.6

Figure 4. Angular segments in the spreadsheet.

2.2 Angular Segments in the Excel File

Each angular segment starts with a header row as shown in figure 5. In this figure, the fifth angular segment header is highlighted (on row 7 of the spreadsheet). The header defines the

segment's lower, upper, and average angle (the first three numbers). The next three numbers define the fragment velocities at the beginning, end, and center of this angular segment.

The last number in the header row (seen in column G) defines the number of weight classes to follow the header row. Note that the first 4 angular segments (rows 3 through 6) contained zero weight classes, i.e., no fragments were found in those angular zones. The first row following the header defines the series of weight classes used to describe the fragments in this angular segment. The next row gives the average number of fragments in each of the weight classes.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	X.1	X.2	X.3	X.4	X.5	X.6	X.7	X.8	X.9	X.10	X.11	X.12	X.13	X.14	X.15
2	36														
3	0	5	2.5		0	0	0	0							
4	5	10	7.5		0	0	0	0							
5	10	15	12.5		0	0	0	0							
6	15	20	17.5		0	0	1293.55	0							
7	20	25	22.5	1293.55	2587.11	2570.71	2								
8	1.55	1.65													
9	0.03	0.04													
10	25	30	27.5	2570.71	2554.32	2558.59	14								

Figure 5. A highlighted angular segment record header.

Examining figure 5, we see that the 20- to 25-degree angular segment header is identified on spreadsheet row 7. Two weight classes are identified in column G of row 7.

For example, in rows 8 and 9, we see that column A holds the first weight class. This class has an average of 0.03 fragment that is in the 1.55-grain weight class. The second weight class is seen in column B where we see 0.04 mean fragment in the 1.65-grain weight class.

2.3 The Shaping Factor

A shaping factor follows in the next row immediately after the data for the last angular segment. Shape factor defines the relationship between the fragments' area and weight averaged over a representative sample of warhead fragments. The definition of shape factor and a description of how it is computed are given in Army FM101-51-3, Joint Munitions Effective Manual, Testing and Data Reduction Procedure for High-Explosive Munitions, chapter 2 (3).

The shape factor is a single number in column A. It is found in the next spreadsheet row following the last angular segment record. Its units by default are in square feet one third power grains per pound

$$\left(\frac{ft^2 \cdot \sqrt[3]{grains}}{pounds} \right).$$

The shaping factor may be optionally converted to grains per cubic inch:

$$\left(\frac{\text{grains}}{\text{in}^3} \right)$$

via the SHAPING_FACTOR_CONVERT configuration file option, as described in the configuration file description (section 6).

3. The CSV and Excel Fragmentation File Schema

Figure 6 is a condensed description of the fragmentation file schema. The descriptions and fields are the same for both file formats (excel and the .csv files).

```

Fragmentation File: FRAGHEADER * FRAGBODY

FRAGHEADER: 1-row of L text fields
FRAGBODY:
    (N: the number of angular segments)
    { angular segment record 1 }
    { angular segment record 2 }
    { angular segment record 3 }
    ...
    { angular segment record N }
    (shaping factor)

angular segment record:
    ( angular segment header )
    ( weight classes )

angular segment header:
    min angle, max angle, avg angle, begin velocity, end velocity, center. velocity, M: the
number of weight classes (M <= L)

weight classes:
    weight class 1, weight class 2, ... weight class M.
    Avg frags in class 1, Avg frags in class 2, ... Avg frags in class M.

shaping factor:
    number

```

Figure 6. Fragmentation file format description.

4. Csv_to_Zdata the Translator

Csv_to_Zdata can be used to convert an Excel fragmentation file into a Z-data file. The Excel spreadsheet file must first be converted to a comma-separated values format file (.csv) by the “save as” feature common to many spreadsheet applications. Csv_to_Zdata can then understand the fragmentation file fields and generate a Z-data file. This process is portrayed in figure 1. Figure 7 displays how the spreadsheet shown in figure 4 would appear after it has been saved as

a .csv file when viewed with a standard text editor. This divides the spreadsheet columns into comma-separated ASCII fields (2).

```

Session Edit View Bookmarks Settings Help
"X.1","X.2","X.3","X.4","X.5","X.6","X.7","X.8","X.9","X.10","X.11","X.12","X.13","X.14","X.15"
36,,,,,,,,,,,,,,,,,,,,,,,,,,,,
0,5,2.5,0,0,0,0,,,,,,,,,,,,,
5,10,7.5,0,0,0,0,,,,,,,,,,,,,
10,15,12.5,0,0,0,0,,,,,,,,,,,,,
15,20,17.5,0,0,1293.55403938858,0,,,,,,,,,,,,,
20,25,22.5,1293.55403938858,2587.10807877715,2570.71245982086,2,,,,,,,,,,,,,
1.55,1.65,,,,,,,,,,,,,,,,,,,,,
0.0272637557541932,0.036677104035983,,,,,,,,,,,,,
25,30,27.5,2570.71245982086,2554.31684086456,2558.59176756262,14,,,,,,,,,,,,,
0.25,0.35,0.45,0.55,0.65,1.05,1.15,1.25,1.35,1.45,1.55,1.65,1.75,3,,,,,,,,,,,,,
0.131325197087032,0.0922049225309656,0.0637423016176277,0.0434689358853011,0.0299638587525028,0
30,35,32.5,2558.59176756262,2562.86669426067,2530.84078230365,11,,,,,,,,,,,,,
0.25,0.35,0.45,1.15,1.25,1.35,1.45,1.55,1.65,1.75,3,,,,,,,,,,,,,
0.06899170376707,0.0484398641100552,0.033487023725692,0.029401347662476,0.0457735816044099,0.07
35,40,37.5,2530.84078230365,2498.81487034663,2297.87227574392,24,,,,,,,,,,,,,
0.25,0.35,0.45,0.55,0.65,0.75,0.85,0.95,1.05,1.15,1.25,1.35,1.45,1.55,1.65,1.75,1.85,1.95,2.05

```

Figure 7. Fragmentation file in .csv format.

4.1 Compiling Csv_to_Zdata

To compile the program, simply move to the Csv_to_Zdata directory and execute the “make” utility from the command line. The Csv_to_Zdata.exe executable program will be created.

Csv_to_Zdata is written in standard C and developed on the Linux operating system. The main directory hierarchy is displayed in figure 8. When the source code is compiled, a GNU³ “makefile” uses this directory organization to first create the support libraries (in libsrc). The CommaDelim subdirectory holds the source code for the “cdf” library. This library is described further in appendix A. Scanner holds a text input scanning library. Csv_to_Zdata’s source code is located in the main Csv_to_Zdata directory. The library functions of scanner and cdf have been previously documented (4). Documentation for them is also available on line under their subdirectories. The Csv_to_Zdata function calls have been documented with Doxygen⁴ (5).

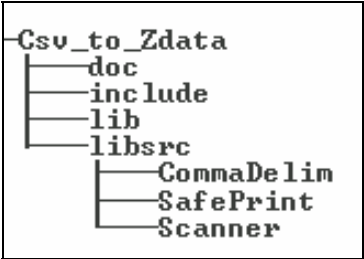


Figure 8. Csv_to_Zdata software directory organization.

³GNU = GNU, not Unix

⁴Doxygen is a software documentation system for C++, C, Java and other languages. It is used much like the documentation system distributed with the Java programming language called Javadoc. Comments may contain “tags” directly in the code. These tags are used to auto-generate some of the software documentation. (Java is a registered trademark of Sun Microsystems.)

Doxygen auto-generated descriptions of these function calls are also available on line and may be viewed in the “doc” subdirectory (Csv_to_Zdata/doc/html/index.html) via a standard web browser.

4.2 Running Csv_to_Zdata Interactively

Cvs_to_Zdata.exe is a command line application that may be run interactively or in batch mode. If no command line options are provided, then by default, the program runs interactively, querying the user for input that determines the configuration of parameters for the translation.

4.3 Running Csv_to_Zdata in Batch Mode

```
Usage:  ./Csv_to_Zdata.exe [-c Configuration_File ] \
        [ -i InputFile.csv ] [ -o OutputFile.ZDATA ]
```

In the author’s opinion, it is much easier to edit preferences in the configuration file (see section 6) with a spreadsheet application and then run Csv_to_Zdata in batch mode. In batch mode, the command line options seen above are

- -c Configuration_File
- -i InputFile.csv
- -o OutputFile.ZDATA

in which

Configuration_File is the name of a configuration file that controls how Csv_to_Zdata translates the fragmentation file.

InputFile.csv is the name of the Excel fragmentation profile spreadsheet that has been saved as a .csv file.

OutputFile.ZDATA is the name you choose for your fragmentation file translated into ZDATA format.

5. Translation Rules

With a few exceptions, the program essentially takes the .csv fields and converts them to a fixed length floating point format that is understood by the FORTRAN (Formula Translator) language used to code CASRED. The exceptions are the optional conversions alluded to and some minor rules:

1. When there are no weight classes following an angular segment record header, at least one class is inserted into the output file anyhow. This is because the FORTRAN source expects to

see at least one class. In this case, the class contains zero fragments and has no effect on the CASRED output.

2. The mass of pentolite that would have the same blast effects as the warhead is the last number in the file. The ability of CASRED to model blast injuries is currently not being used so this last number is normally entered as (0.000).
3. A maximum of seven weight classes or fragment numbers can be written on any one line. If the number of weight classes in an angular segment exceeds seven, then additional lines of weight class and additional lines of fragment numbers must be used. For example, if the last number in the angular segment head files list 19 weight classes, then there will be three lines of weight class followed by three lines of fragment numbers. The first two lines of weight class will contain seven class each and the third line will contain five. The fourth and fifth lines following the header will contain seven fragment numbers each. The sixth line will again contain five numbers. If the number of weight classes is smaller than seven, there will be a single line of weight class and single line of fragment numbers. The entries on the line will equal the actual number of weight classes.
4. The configuration tracking data are appended to the Z-data file as additional information. This includes the date and time of the translation, the name of the input file, the output (Z-data) file, and who created the file. Additionally, the configuration parameters used to convert the file are appended. These configuration parameters are appended in .csv format so that they can actually be used verbatim as the configuration file for future translations. All this appended information is ignored by CASRED because the FORTRAN code stops reading before this appended section. Figure 9 displays the first part of the Z-data output file that was translated from the input shown in figure 5.

Angular Segment	Weight Class 1	Weight Class 2	Weight Class 3	Weight Class 4	Weight Class 5	Weight Class 6	Weight Class 7
zdata	36.000						
0.000	5.000	2.500	0.000	0.000	0.000	0.000	1.000
1.000							
0.000							
5.000	10.000	7.500	0.000	0.000	0.000	0.000	1.000
1.000							
0.000							
10.000	15.000	12.500	0.000	0.000	0.000	0.000	1.000
1.000							
0.000							
15.000	20.000	17.500	0.000	0.000	1293.554		1.000
1.000							
0.000							
20.000	25.000	22.500	1293.554	2587.108	2570.712		2.000
1.550	1.650						
0.027	0.037						

Figure 9. A Z-data file after a successful translation (a fixed length ASCII record format).

Comparing the .csv input in figure 5 with translated output in figure 9, we note that the first four angular segments (rows 3 through 6 in figure 5) contained no weight class, but by conversion rule 1, one “empty” weight class has been added for each of these segments in the Z-data file.

6. Csv_to_Zdata Configuration File and Parameters

The configuration file may be used to tailor translation options and change the default units in the translated output. Default unit measurements found in the fragmentation profile are listed in table 1.

Table 1. Default units of measure.

Measure	Units	Unit Conversion Factor (found in the Csv_to_Zdata configuration file)
Angles	degrees	CONVERSION_FACTOR_ANGLES
Mass	grains	CONVERSION_FACTOR_MASS
Velocity	feet per second	CONVERSION_FACTOR_VELOCITY

The configuration file is a .csv text file and may be edited with a standard text editor or a spreadsheet. The sample file shown in figure 10 is displayed with a spreadsheet. This configuration file contains a series of parameter names and their values. Parameter names may appear in any order. The parameters and their effects are explained. Parameters not appearing in the configuration file take default values.

	A	B	C
1	<u>ParamName</u>	<u>ParamValue</u>	Comment
2	MUNITION_NAME	New Grenade	name of the munition
3	CLASSIFICATION_NAME	unclassified	security classification of this information
4	OUTPUT_FORMAT_NAME	zdata	should not be changed: output format name
5	OUTPUT_FORMAT_NUMERIC_FIXED	%10.3f	should not be changed: Defines the fixed field length of the output values
6	SHAPING_FACTOR_CONVERT	0	can be one of: 0,1,2. Meaning of each [0: no change], [1: convert from ((ft^2) (grain^(1/3)))/lbs) to (grains/in^3)], [2: inverse of 1]
7	CONVERSION_FACTOR_ANGLES	1	May be used to convert angular units. Default angles are in degrees.
8	CONVERSION_FACTOR_VELOCITY	1	May be used to convert velocity units. Default velocity is ft/sec.
9	CONVERSION_FACTOR_MASS	1	May be used to convert mass units. Default mass are grains.

Figure 10. Csv_to_Zdata configuration file. (The header row and comments column are both ignored.)

6.1 Munition_Name

This is a text field to identify the munition. It has no effect other than being echoed in the output file (seen on the second line of figure 10).

6.2 Classification_Name

This is a text field to identify the security classification of the data. It has no effect other than being echoed in the output file (seen on the first line of figure 10. This is not intended to conform to classified information markings as required by U.S. Army regulations.

6.3 Output_Format_Name

This parameter identifies the output format scheme; “zdata” is used here. This field is ignored by the FORTRAN code.

6.4 Output_Format_Numeric_Fixed

This value (10.3F) should definitely not be changed since the FORTRAN READ statements are expecting fixed length floating point fields of 10 digits with three decimal places.

6.5 Shaping_Factor_Convert

The shaping factor conversion flag: By default, the shaping factor is in (A) square feet one third power grains per pound $\{[(\text{ft}^2) (\text{grain}^{(1/3)})/\text{lb}]\}$. This may be optionally converted to (B) grains per cubic inches (grains/in³). If this flag is set to 0, then no conversion occurs to the shaping factor. If this flag is set to 1, it will convert the shaping factor value from (A) to (B) units. If this flag is set to 2, it will convert from (B) to (A) units.

6.6 Conversion_Factor_Angles

By default, angles are in degrees. All angular measures written to the translated output are multiplied by CONVERSION_FACTOR_ANGLES. Thus, if the spreadsheet angle is in degrees, leave CONVERSION_FACTOR_ANGLES set to 1.0. On the other hand if the data in the spreadsheet are in radians, set CONVERSION_FACTOR_ANGLES to 57.295777951 and convert to degrees (i.e., 180 degrees/ π radians).

6.7 Conversion_Factor_Velocity

By default, velocity is feet per second. All velocity measurements written to the translated output are multiplied by CONVERSION_FACTOR_VELOCITY. Leaving CONVERSION_FACTOR_VELOCITY set to 1.0 will keep the default feet-per-seconds units. If the spreadsheet uses meters per second for fragment velocities, convert these velocities to feet per second by using a CONVERSION_FACTOR_VELOCITY equal to 3.2800840.

6.8 Conversion_Factor_Mass

By default, mass is in grains. All mass measurements written to the translated output are multiplied by CONVERSION_FACTOR_MASS. If the spreadsheet uses grains for fragment weight, leave

CONVERSION_FACTOR_MASS set to 1.0. If the spreadsheet uses kilograms for fragment weight, convert to grains by using a CONVERSION_FACTOR_MASS of 15432.3583529 and if the spreadsheet fragment weights are in grams, use 15.432358.

7. Summary

CASRED is a computer model that computes probability of incapacitation as a result of a round or fragment impact for an individual (human) target with or without ballistic protection. For fragment threats, CASRED requires a description of the fragment masses, velocities, and distribution densities. These fragment patterns can be described as an ASCII file in Z-data format or as spreadsheet in JMEM (Joint Munitions Effectiveness Manual) format.

8. References

1. Oberle, W.F.; Anderson, R.D. *Sensitivity of Casualty Reduction (CASRED) Results Used in Weapon Effectiveness Monte Carlo Simulations*; ARL-TR-3762; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, April 2006.
2. Shafranovich, Y. Common Format and MIME Type for Comma-Separated Values (CSV) Files, Request for Comments: 4180, October 2005, <http://tools.ietf.org/html/4180>.
3. U.S. Army. *Joint Munitions Effective Manual, Testing and Data Reduction Procedure for High-Explosive Munitions, Revision 2*; FM101-51-3; U.S. Army Materiel System Analysis Activity: Aberdeen Proving Ground, MD, May 1989.
4. Sauerborn, G.C. *The Distributed Interactive Simulation (DIS) Lethality Server Volume II: User and Programmer's Manual*; ARL-TR-1775, page 63; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, February 1999.
5. Dimitri van Heesch, "Doxygen Manual", latest release v1.4.7 - last update 11 Jun 2006 by Dimitri van Heesch. Available at <http://freshmeat.net/projects/doxygen>.

Appendix A. Comma-Separated Values (CSV) Library

This appendix describes the interface functions of the library used to parse the CSV records. The library is known as the comma-delineated field (cdf), first published in 1999 (4). Since then, a practical specification has been published standardizing the CSV format (2). In particular, this CSV specification clarifies the literal quote character that was ambiguous in 1999.

Functions

- char * **cdf_strtok** (char *str, int *flag, int literal)
- int **cdf_scan_fields** (char *fields[], int num_fields, char *buffer, int *cdfflag)

Function Descriptions

- int **cdf_scan_fields** (char * fields[], int num_fields, char * buffer, int * cdfflag)

```
~ cdf_scan_fields()
*
* int cdf_scan_fields(char *fields[],int num_fields,char*buffer,int *cdfflag)
*
* cdf_scan_fields() scans the comma separated fields found in "buffer"
*                   the first "num_fields" fields are read and
*                   malloc'd duplicates of these strings are returned
*                   in the array "char *fields[]".
*
*
* Returns 0 on error.
* Otherwise the number of fields read and returned in "fields[]"
* is returned.
* if a quoted comma separated field ends prematurely,
* then -1 is returned in cdfflag.
*
```

- char* **cdf_strtok** (char * str, int * flag, int literal)

```
~ cdf_strtok()
*
* char *cdf_strtok( char *str , int *flag, int literal );
*
* Similar to strtok(3) but specialized for comma-delimited fields.
* cdf_strtok() reads and returns pointers to the comma separated
* field(s) within the string "str". The returned string will be
* the characters found between field separators (the comma).
* [Note: If the field is empty, (e.g. ",," then an empty string
* would be returned (i.e. "").]
*
* Subsequent calls will (with str=NULL) will return subsequent
* comma-dilimated fields in the original string until no more
* fields are found, (and a NULL is then returned).
*
*
* flag is set to -1 if the field ended
* before a comma or newline was found (this can only occur
* in the case of a string field. (A quoted string field).
```

```
*      That is when (") is the first character in a field,
*      (then we expect a string field)
*      in which case the string contents is read
*      and the whole field content is return
*      (including the enclosing quotes (").
*
*      literal
*      Within a string field the quote character (")
*      can be quoted by placing a the literal character
*      directly in front of the quote character
*      Many databases may used the quote '''
*      (e.g. "This is a string with ""quotes"")
*      Also the literal character is often the backslash '\\
*      (e.g. "This is a string with \"quotes\"")
*
*      Geoff Sauerborn
*      U.S. Army Research Laboratory
*      geoffs@arl.army.mil
*      http://www.arl.army.mil/
*
```

NO. OF
COPIES ORGANIZATION

- 1 DEFENSE TECHNICAL
(PDF INFORMATION CTR
ONLY) DTIC OCA
8725 JOHN J KINGMAN RD
STE 0944
FORT BELVOIR VA 22060-6218
- 1 US ARMY RSRCH DEV & ENGRG CMD
SYSTEMS OF SYSTEMS
INTEGRATION
AMSRD SS T
6000 6TH ST STE 100
FORT BELVOIR VA 22060-5608
- 1 DIRECTOR
US ARMY RESEARCH LAB
IMNE ALC IMS
2800 POWDER MILL RD
ADELPHI MD 20783-1197
- 1 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL CI OK TL
2800 POWDER MILL RD
ADELPHI MD 20783-1197
- 2 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL CS OK T
2800 POWDER MILL RD
ADELPHI MD 20783-1197
- ABERDEEN PROVING GROUND
- 1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL CI OK (TECH LIB)
BLDG 4600
- 1 US ARMY EDGEWOOD RDEC
ATTN SCBRD TD J VERVIER
APG MD 21010-5423
- 6 US ARMY RESEARCH LABORATORY
ATTN AMSRD ARL WM BF
BLDG 309 S WILKERSON
C PATTERSON
R PEARSON
G SAUERBORN (3)
- 1 US ARMY RESEARCH LABORATORY
ATTN AMSRD ARL SL BE
W MERMAGEN, JR
BLDG 390A