

BioWare Aurora Engine

2DA File Format

1. Introduction

A 2da file is a plain-text file that describes a 2-dimensional array of data.

In BioWare's games, 2da files serve many purposes, and are often crucial to the proper functioning of the game and tools. They describe many aspects of the rules and game engine.

Although 2da files are plain text files, they are structured according to a set of rules that must be followed in order for the game and tools to read them correctly.

2. General Concepts

The main body of a 2da file is a table containing rows and columns of data. Each individual data element at a given row/column coordinate is called an entry. The data may be text, integer, or floating point values.

Consider the following example of the contents of a 2da file:

2DA	V2.0				
	LABEL	STREF	STRING	HasLegs	Personal_Space
0	Chicken	2013	Chicken	1	0.13
1	****	****	****	****	****
2	Battle_Horror	1996	"Battle Horror"	0	0.3
3	Bear_Polar	1999	"Polar Bear"	1	0.6
4	Deer	2017	Deer	1	0.6

The above example illustrates several points elaborated on below.

Whitespace separating columns

Each column is separated by one or more spaces. The exact number of spaces does not matter, so long as there is at least one space character. The columns do not have to line up exactly, as shown by row 3 in the example above.

Important: do not use tab characters to separate columns.

First column

The first column always contains the row number, with the first row being numbered 0, and all subsequent rows incrementing upward from there.

The first column is the only column that does not have a heading.

Note that the numbering in the first column is for the convenience of the person reading or editing the 2da file. The game and tools automatically keep track of the index of each row, so if a row is numbered incorrectly, the game and tools will still use the correct number for the row index. Nevertheless, it is a good habit to make sure that rows are numbered correctly to avoid confusion.

Column names

All columns after the first one must have a heading. The heading can be in upper or lower case letters and may contain underscores.

Data types

There are three types of data that may be present in a 2da. All data under a given column must be of the same type. The data types are:

- **String:** a string can be any arbitrary sequence of characters. However, if the string contains spaces, then it must be enclosed by quotation mark characters (") because otherwise, the text after the space will be considered to belong to the next column. The string itself can never contain a quotation mark.
- **Integer:** an integer can be up to 32-bits in size, although the application reading the integer entry is free to assume that the value is actually of a smaller type. For example, boolean values are stored in a 2da as integers, so the column for a boolean property should only contain 0s or 1s.
- **Float:** a 32-bit floating point value.

The 2da format does not include data type information for each column because the application that reads the data from the 2da already knows what datatype to assume each column contains.

Blank (****) entries

The special character sequence **** indicates that an entry contains no data, or the value is not applicable. Note that this character sequence contains *exactly* 4 asterisk characters, no more and no less.

When deleting a row from a 2da file, all columns in that row should be filled with ****s.

The **** value is also used to indicate "N/A".

An attempt to read a String from a **** entry should return an empty string (""). An attempt to read an Integer or Float should return 0. The programming function that performed the reading operation should indicate that the read attempt failed so that that application knows that the entry value is no ordinary "" or 0.

StrRefs

One common use of Integer columns is to store StringRefs (or StrRefs). A StrRef is an index into the user's dialog.tlk file, which contains strings in the user's native language. When a 2da file includes information that relates to text that needs to be displayed to the user, that text is not embedded directly in the 2da file itself. Instead, the 2da contains the StrRef for the text.

Using StrRefs in a 2da allows all languages to use the same copy of a 2da. Instead of providing a few hundred different 2das for each language, it is only necessary to change a single file, dialog.tlk.

3. File Layout

Line 1 - file format version

The first line of a 2da file describes the version of the 2da format followed by the 2da file. The current version header at the time of this writing is:

```
2DA V2.0
```

Line 2 - blank or optional default

The second line of a 2da file is usually empty.

Optionally, it can specify a default value for all entries in the file. The syntax is:

```
DEFAULT: <text>
```

where <text> is the default value to use. Note that the default text is subject to the same whitespace rules as any other column in a 2da. A string containing spaces must therefore be enclosed by quotation marks.

The default value will be returned when a requested row and column has no data, such as when asking for data from a row that does not exist. For String requests, the default text is returned as a string. For Integer or Floating point requests, the default will be converted to an Integer or Floating point value as appropriate. If the default string cannot be converted to a numerical type, the return value will be 0. The programming function that reads the 2da entry should indicate that the read attempt failed.

The default value is *not* returned when a requested entry is ****. An entry that contains **** will return a blank string or zero.

Line 3 - column names

The third line of a 2da file contains the names of each column. Each column name is separated from the others by one or more space characters. The exact number of spaces does not matter, so long as there is at least one.

A column name contains alphanumeric characters or underscores, and can begin with any of these characters (ie., not restricted to starting with a letter).

Lines 4 to infinity - row data

All lines after and including line 4 are data rows.

Each column in a row is separated from the other columns by one or more space characters. When viewing the contents of a 2da using a fixed-width font, the columns in each row do *not* have to visually line up with the columns in the other rows, but for ease of reading, it is best to line them up anyway.

The very first column in a row is the row's index. The first data row (line 4) has an index of 0, the second data row (line 5) has an index of 1, and so on.

Every row must contain the exact same number of columns as there are column names given in line 3, plus one (since the index column has no name).

If the data for a column is a string that contains spaces, then the data for that column should begin with a quotation mark and end with a quotation mark. Otherwise, the text after the space will be considered to belong to the next column. Because of how quotation marks are handled, a string entry in a 2da can never contain actually quotation marks itself.

4. Maintenance

After a 2da file has been created and support for it has been added to the game and/or tools, it will often be necessary to make changes to the 2da. The following rules govern how to safely make changes.

Columns

Applications may reference a column by position (column 0, column 1, etc.) or by name. To avoid breaking code that depends on column position, the following rules apply:

- *Always* add new columns after the very last column.
- *Never* insert a new column inbetween two existing ones or as the first one.
- *Never* delete a column from a 2da.
- *Never* rename a column.
- When adding a column, make sure that all rows include entries for the new column.

Rows

Many game object properties are integer values that serve as indices into particular 2da files. Consequently, care must be taken when changing 2da row data to ensure that a minimum amount of existing data is affected by the change.

Always add rows to the very end of the file.

Never insert a row inbetween two other existing rows.

Never delete a row. If it is necessary to remove the data in a row, fill the row with * * * * entries instead. Try to ensure that no existing data, in a 2da or otherwise, references the starred-out row.