



E. A. G. L. E. Euphoria Adventure Game Language Engine Vers 0.01alpha

(C)opyright 1998,1999,2002 Shadetree Software & Ferlin Scarborough

November 1998 - February 2002

Email: ferlin@sandw.net or ferlin_s@yahoo.com

* W A R N I N G * W A R N I N G * W A R N I N G * W A R N I N G *

Very LOOOOONG winded Author Babble Ahead, for the quick tour, just skip to the *You May Start Here* Label, or just read on, if you have some time.

* W A R N I N G * W A R N I N G * W A R N I N G * W A R N I N G *

What is **EAGLE** you ask? Well, in order to answer that we need to take a trip back in time a few years. Let's strap our selves into the Virtual Time Ship (VTS), turn the Time Dial to 1985 and press the LAUNCH button.

With a tiny little shutter the VTS proceeds to traverse the dimensions of Time, 1998 1997 1996 1995 and so on until it arrives at it's destination of 1985.

We find our selves looking over the shoulder of a 20 year old man sitting in a bedroom staring at a Color TV he has hooked up to a TRS-80 Color Computer, 64K with Extended Basic. The young man is getting bored with losing to the computer in Black Jack and Chess, looking at the books that came with the computer he sees an Extended Basic Manual, Hmmmm I wonder he says to him self? He starts to read the book the clock reads 2:00pm, he starts to peck at the keyboard, and by 3:30am he has modified the programs of Black Jack and Chess with Cheat Codes so he can WIN anytime he likes.

Needless to say MANY sleepless nights later he has started to write little programs for some of his friends. He discovers that he LOVES to control the machine, and decides to take a computer course, during this time he buys him self an IBM PC with a Blazing Speed of 4.77 MHz, 10 Meg hard drive, 1200 Baud internal modem and 1 Meg of Ram (One of the BEST setups in the neighborhood.)

One of his class mates gives him a copy of a program called Hitchhikers Guide to the Galaxy, it has NO fancy graphics to show off his EGA monitor, but still there is something intriguing about it. An all text adventure game, forcing him to use his imagination, suddenly he is hooked on text adventure games, and spends hours trying to figure out what he needs to do to get passed the Troll or open the locked door. The young man decides

he wants to write his own adventure game, but he gets involved with going to work in the computer industry and eventually puts the Adventure Game Writing on the back burner, where it just simmers for the next few years.

The young man grows older, but the dream of creating his own Adventure Game never fully leaves his mind, he discovers a Tool Kit called AGT, and starts to design his own game, but gets frustrated with trying to get AGT to do some things it was not designed to do, a couple more years go by and he try's a programming language called INFORM, and another called TADS, and another called ALAN, in the end he decides his dream will not be met.

One day he discovers a language called **EUPHORIA**, it was simple to learn, and there was a great bunch of folks on a Listserv, that helps him through some rough spots, he starts to write a few small programs, and decides the only way he will EVER make his DREAM happen is to design his own Game Engine, but he is just learning **EUPHORIA**, and is not sure how to take the input from a user and turn it into a Procedure or Function Name, he starts to get discouraged again. Then it happens, one day he reads on the Listserv about a program called EuServe, downloads it and takes a gander at the source code, BINGO why didn't HE think of that.

The young man starts to write code and **EAGLE** is the results of this LONG trip, thanks to **EUPHORIA** and the wonderful folks in the **Euphoria** Community.

You May Start Here

WHEWWWWW!, got a little carried away there, but the point is that **EAGLE** has been YEARS in the making, and it is just now starting to come to life, and I am a bit enthused with it.

Standard Disclaimer

This program is being distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

I Ferlin Scarborough or **Shadetree Software** do not accept responsibility for any effects, adverse or otherwise, that this code may have on you, your computer, your Mental or Emotional Health, and so on.

USE IT AT YOUR OWN RISK.

This Software MAY include some Trade Secrets of **Shadetree Software**, and at this time the Author does not wish to release the SOURCE CODE, so PLEASE DO NOT ask for it, However the Author does reserve the right to release such CODE into the Public Domain at any time in the future (Which there is a VERY GOOD chance this may occur.)

PLEASE out of Respect for the Author and the time and effort he has placed into the design and coding of this software package. DO NOT modify, adapt, translate, create derivative works, decompile, disassemble, or otherwise reverse engineer or derive source code from, all or any portion of the software or anything incorporated therein or permit or encourage any third

party to do so.

NOTICE - if you have any modifications or enhancements you would like to see made to this software package, PLEASE contact the Author with these ideas, so that YOU can receive Credit for these changes. Refer to the Questions, Comments, etc section later in this document for information on how to reach the Author or Shadetree Software.

The Author will review and consider ALL reasonable suggestions with an open and receptive mind.

The Author reserves the right to have FINAL decision in what changes will be made to his product.

**Let's give Credit where Credit is Due:
And the Awards go to:**

The following list does not by any means include ALL those that made this system possible, I want everyone to know ahead of time, that if I left anyone out it was unintentional.

This program is being brought to you in part by:

Hawke (A.K.A Mike De Land) - THANKS for the inspiring EuServe!
Ad Reinks
Wolf Fritz
David Cuny
David Gay (Whose GREAT tutorials ABGTE and ABGTE2 taught me the beginnings)
Bernie Ryan (Thanks for replacing my lost copy of EUSERVE)
Ralf Nieuwenhuijsen (Hope the Last Name is spelled right?)
Jiri Babor
Greg Harris
The **Euphoria** Community on the List Serve
(<http://www.topica.com/lists/EUforum/read>)
RDS Software and Robert Craig (Father of **EUPHORIA**)

Any others I may have missed, many have unknowingly contributed to this game system.

EUPHORIA - the language this system is written in, has a public domain version available at <http://www.RapidEuphoria.com/>
A registered version is available CHEAP (about \$39.00 at the time of this compilation.)

**What features does EAGLE have?
Now Showing - Tonight Only!**

To begin with I think it should be made clear here that **EAGLE** is **not** by any means a completed work.

This version (0.01 Pre-Alpha) is being released solely with the purpose of giving someone in the General Public a chance to run it and see what they think of it so far, also to have them give any ideas or comments on how they believe **EAGLE** should proceed from here.

So far all there is to **EAGLE** is a compile program and a run time engine, later in this documentation is a list of things to be added to **EAGLE** before it is released for actual use. A brief description of the compiler and the run-time Engine follows.

Compiler:

The compiler so far only does a few basic functions, it takes a game source file (.EGL) and builds it into a database file for the game. The game source file has just a few things in it right now. There is no Error Checking, so BEWARE if you modify the demo EGL file, the programs may no longer function.

Let's break down the EGL file and see what it is made of:

The EGL Source File contains the following **GAME TOKENS** and **TOKEN PAIRS**:

[**TITLE**] .. [**END_TITLE**] The lines between these TOKENS are the Games Title.

[**INTRO**] .. [**END_INTRO**] The lines between these TOKENS are the Introduction.

[**INSTRUCT**] .. [**END_INSTRUCT**] Between these TOKENS are the Instructions, to be shown if Player answers Y to Instructions.

[**START_LOCATION**] The Location (Room) where the Game is to Start.
(i.e. [**START_LOCATION**] **Bed_Room**) tells the compiler that the player is to be placed in the Bed Room at the start of the game.

[**MAX_SCORE**] The MAXIMUM Score a Player can achieve. (i.e. [**MAX_SCORE**] **1000**) tells the compiler that the Maximum Score a Player can reach is **1000**.

[**MAX_WEIGHT**] The MAXIMUM Weight a Player can carry. (i.e. [**MAX_WEIGHT**] **100**) tells the compiler that the Maximum Weight a Player can carry in his/her inventory is **100**.

[**LOCATION**] .. [**END_LOCATION**] Between these TOKENS is Location (Room) specific information.

[**DESCRIBE**] .. [**END_DESCRIBE**] Between these TOKENS is a Description of the named Location.

[**OBJECT**] .. [**END_OBJECT**] Between these TOKENS is Object (Item) specific information.

[**OBJECT_DESCRIBE**] .. [**END_OBJECT_DESCRIBE**] Between these TOKENS is a description of the named

Object (Item).

[NPC] .. [END_NPC] Between these TOKENS is NPC (Non Player Characters) specific information.

[NPC_DESCRIBE] .. [END_NPC_DESCRIBE] Between these TOKENS is a description of the named NPC (Non Player Character)

Templates for the 3 Main TOKEN Structures LOCATION - OBJECT - NPC

Below are the Templates for the 3 Main TOKEN Structures used in an **EAGLE** Adventure.

LOCATIONS TEMPLATE (Example):

```
[LOCATION] Drawing Room West      <-- Name of the Location.
  Short_Name: Drawing Room      <-- This is the name displayed on the status
                                line.
  East: Drawing Room East       <-- Direction and location that direction
                                leads to. Available Directions are
                                North:, South:, East:, West:, NorthEast:,
                                NorthWest:, SouthEast:, SouthWest:, Up:,
                                Down:, Enter: and Exit:
  End_Game:                     <-- This causes the Game to End if the player
                                enters this room.
  AddPoints: 12                 <-- Adds number of points to players score, in
                                this case it would add 12 points to score.
  SubPoints: 10                 <-- Subtracts number of points from players
                                score, in this case it would subtract 10
                                points from the players score.
  Win_Game:                     <-- Player Wins the game when they enter this
                                room.
[END_LOCATION] Drawing Room West <-- End of Location Specific Information.
```

The Location Specific Information **MUST** be followed by a Location Description this is the description used when the player LOOKS at the Location.

```
[DESCRIBE] Drawing Room West    <-- Name of the Location.
  You are standing on the west side of the fireplace in the drawing room of
  your house. The chairs here are made of nicely polished wood, and the mantle
  over the hearth is very well kept up. Everything, in fact, is exactly as it
  should be. The sun is shining, birds are singing, and you feel that today of
  all days, nothing could possibly go wrong.
[END_DESCRIBE] Drawing Room West <-- End of Location Description.
```

OBJECTS TEMPLATE (Example):

```

[OBJECT] Tangled Yarn      <-- Name of the Object (Item).

  Short_Name: Yarn        <-- Short Name, another name the player can refer
                           to this Object (Item) by.
  Describe: There is a ball of yarn lying on the floor here <-- This will
                           be the description that is displayed when the
                           Location description is given to the player.
  Location: Drawing Room West <-- The Location that the Object (Item)
                           starts out in at beginning of game.
  Inventory: N            <-- This is either an N or Y, telling if this
                           Object (Item) is in the players Inventory.
                           If this is a Y, this Object will be placed
                           in the players Inventory at beginning of game.
  Weight: 1              <-- Weight of the Object, this is used to determine
                           if the player can carry it in inventory or not
                           depending on what the MAX_WEIGHT is set to.
  AddPoints: 5          <-- This works the same way as AddPoints does for
                           a Location, it will Add number of points to
                           the players score, in this case 5 points would
                           be Added when the player picked up this Object.
                           If this is 0, then NO points are added.
  SubPoints: 5          <-- This works just like AddPoints only it Subtracts
                           the number of points from the players score
                           rather than Adds it. In this case it would
                           Subtract 5 points from the players score when
                           they picked up this Object.
  IsContainer:          <-- This Object is a Container, so other Objects
                           can be placed inside of it. To place another
                           Object inside this one, just set that Objects
                           Location: label equal to the Name of this
                           Object.
  Closed:               <-- If Object is a Container, this tells the
                           compiler that it starts out Closed.
  Open:                 <-- If Object is a Container, this tell the
                           compiler that is starts out Open.
  [READ] Made in China [END_READ] <-- This tells the compiler that this
                           Object is a Readable Object, and the TEXT
                           between [READ] and [END_READ] is what the
                           player will see if they typed in Read Object.
                           Such as in this case read yarn. The player
                           would be told Made in China.

[END_OBJECT] Tangled Yarn <-- End of Object specific information.

```

The Object specific information **MUST** be followed by an Object Description. This is the description used when the player LOOKS at or EXAMINES the Object.

```

[OBJECT_DESCRIBE] Tangled Yarn      <-- Name of the Object (Item).
The ball of yarn seems to be incredibly tangled. In fact, it seems to have
gotten tangled into the rug itself!
[END_OBJECT_DESCRIBE] Tangled Yarn <-- End of Object Description.

```

NPC (Non Player Character) **TEMPLATE** (Example):

```
[NPC] Froobious Bandersnatch <-- Name of Non Player Character.

  Short_Name: Bandersnatch      <-- Short name of character, another name
                                   the player can refer to the character
                                   by.
  Describe:   There is a Froobious Bandersnatch here. <-- This is what will
                                   be displayed to the player when they
                                   enter the Location this character is in.
  Location:  Denser Forest      <-- Location where this character starts out
                                   at the beginning of the game.
```

NOTE: The next Three parameters Weapon, Attempts, and Turns, alert the compiler that this character is a **HOSTILE** and not FRIENDLY character.

```
  Weapon: GOLDEN SWORD          <-- Object (Item) used to KILL or defeat
                                   this character with.
  Attempts: 3                   <-- Number of Attempts (Attacks) the player
                                   can make to Kill this character before
                                   the character kills the player. The
                                   character would kill the player because
                                   of Unsuccessful attempts (i.e. using
                                   wrong weapon).
  Turns: 5                       <-- Number of Turns the player can remain
                                   in the Room with the character before
                                   the character would attack and kill
                                   the player.
```

```
[END_NPC] Froobious Bandersnatch <-- End of Character specific Information.
```

The Non Player Character specific information **MUST** be followed by an NPC Description. This is the descriptions that would be displayed to the player if the LOOKED at or EXAMINED the NPC.

```
[NPC_DESCRIBE] Froobious Bandersnatch <-- Name of Non Player Character.
The Bandersnatch is snorting and drooling. It seems to harbor no very
honorable intentions towards you.
[END_NPC_DESCRIBE] Froobious Bandersnatch <-- End of NPC Description.
```

File Names and Extensions being used by the EAGLE System

ECOMPILE.EXE - The Game Compiler used to compile EGL files.
REAGLE.EXE - The Run Time Engine, used to execute the compiled Games.
EAGLE.DOC - This Documentation file you are currently reading, in Microsoft Word Format.
EAGLE.TXT - The Documentation in Plain Text Format.
EAGLE.MSG - Standard Messages File, contains messages output during

game play (i.e. Sorry but I don't understand what you mean by that?)

- GAMENAME.EGL** - **Eagle** Game Source Code File.
- GAMENAME.MSG** - **Eagle** Game Messages File used to **CUSTOMIZE** the messages output by your game. If not present Compiler uses **EAGLE.MSG** (Standard Messages File.)
- GAMENAME.IGL** - **Eagle** Game Source Code Include File. (*** FUTURE USE ***)
- GAMENAME.MGL** - **Eagle** Game Source Code Special Module Files. (*** FUTURE USE ***)
- GAMENAME.BAT** - A Batch File created to allow the user to enter the Game Name to execute a Game.

The following file is created during Game Compile to contain the Game information to be used by the Run Time Engine.

GAMENAME.EDB - Game Database File created during Game Compile.

The files needed to run a Compiled Game would consist of the following:

- REAGLE.EXE** - The Run Time Engine.
- GAMENAME.BAT** - A Batch File to allow the user to enter the Game Name to Execute the Game.
- GAMENAME.EDB** - The Game Database containing the game information needed.

Future Enhancements, Additions and Planned Direction for EAGLE.

As stated earlier in this documentation **EAGLE** is ***NOT*** a completed work. Although **SOME** of the following are already being developed for the **NEXT** release of **EAGLE**.

Future abilities of the system will include, but not be limited to the following:



Graphics capability.



Compound Sentence Parser in the Run Time Engine.



Better Error handling by both the Compiler and Run Time Engines.



A source code Editor (with many spoiler options).



Sound capability.



Include capability in the Compiler for source code files.



Multiple Language capability, plug in modules for other languages besides English.



Ability for USERS to OVER-RIDE built in Sub-Routines.



Multiple Screen Modes.



Windows Version of both the Compiler and Run Time Engines.



A Fully Functional Command Language.



Hopefully some day there will also be an AI Module for the NPC's (Non Player Characters), this will allow NPC's to have their own Personality.



Initialization Modules for Color, Fonts and so on.....



A Module for Verb and Synonym Substitution, as well as Unique Terms for specific games.

Where to find the Author for Questions, Comments, Etc.

If you have any questions, bug reports, etc., send me an e-mail at:

ferlin@sandw.net or **ferlin_s@yahoo.com** or **shadetreesoft@netscape.net**

For the Latest information on EAGLE and other products produced by Shadetree Software check out our home page at:

<http://mywebpage.netscape.com/shadetreesoft>

Want to know more about what is going on at Shadetree Software and what the

future holds for EAGLE and other Shadetree Products, just sign up for our FREE News Letter while your looking our Website over.

Release History

??/??/???? - First (Pre-Alpha) Release of **EAGLE**.