




Bright-Crayon, LLC  
501 E Franklin Street  Richmond, Virginia 23219  U.S.A.

Partners:  
[www.bright-crayon.com](http://www.bright-crayon.com)  
+1.804.307.2729

George Kelly Flanagan  
Susan R. Jacobs

Dr. Pamela Kiecker  
John N. Pastore, Jr.  
Greg Provo

## Empire Inventory

Compiled by George Flanagan

Revision Number	Date of last Revision	Number of Pages
 7	 200411230950	 7

1.OVERVIEW.....

2.LIBRARY MODULES.....

2.1.CMDTABLE.....

2.2.COORDINATES.....

2.3.DIRECTFILE.H.....

2.4.EMPIREDEBUG.H.....

2.5.EMPIREMATH.....

2.6.EMPIRETYPES.H.....

2.7.GAMETIME.....

2.8.GLOBALS.....

2.9.ISLAND.....

2.10.MATRIX.H.....

2.11.PARSER.....

2.12.PLAYER.....

2.13.PROMPT.....

2.14.SCANNER.....

2.15.SECTOR.....

2.16.SEMANTICS.....

2.17.SHIP.....

2.18.SINGLETON.H.....

2.19.STOPWATCH.....

2.20.TOPLEVEL.H.....

2.21.UTILITIES.....

3.TOP LEVEL MODULES.....

3.1.CLOSE.....

3.2.DUMP.....

3.3.GENESIS.....

3.4.MAIN.....

3.5.OPEN.....

3.6.TEXTMAP.....

3

4

4

4

4

4

4

4

4

5

5

5

5

5

5

5

5

5

5

6

6

6

6

7

7

7

7

7

7

## 1. Overview

---

This document contains a list of the source files that make up the Empire game.

Many of the files consist of a CPP and H pair, and that assumption is the safe one to make in the absence of other information.

## 2. Library Modules

---

### 2.1. Cmdtable

A set of routines necessary to execute the commands.

### 2.2. Coordinates

Empire is played on a large grid that constitutes the “game board,” or “world.” This module contains the many functions necessary to support calculations of distances and directions. It is worth noting that there are actually two frames of reference: island coordinates and world coordinates, both of which are to be found in these files.

### 2.3. Directfile.h

Directfile is a C++ template that allows an ease of manipulation of files that can be construed to be made of fixed length records. Empire's database consists of several of these types of direct access files; including one file each for ships, sectors, islands, and players.

### 2.4. Empiredebug.h

This is a file containing only preprocessor macros that allow a programmer to easily insert statements that are only executed if the global variable “Verbose” is set to true.

### 2.5. Empiremath

This module contains only the special random number generator that is used by the game. One of the special things about it is the ability to explicitly seed the generator in order to reproduce entire games from the same key.

### 2.6. Empiretypes.h

To support both order and ease of debugging, Empire makes use of the “enum” type in the C++ language for types of sectors and materiel.

### 2.7. Gametime

The game's clock always starts at zero (the beginning of the game), and runs to some point at which the game is over (often a number around 2000). These routines to convert actual time into game time, and the reverse.

### 2.8. Globals

Each Empire game has a number of “laws of the universe” that apply to the current game: length of a tick on the game's clock in actual time, the names of and pointers to the data files of the game, etc. There are also other constants that apply to just about any game, such as ship parameters. All of these are to be found in this module.

The Globals structure is intended to be used within the Singleton construct mentioned below.

## 2.9. Island

An Island is a record in the Island file. The record contains such things as the island's size, its exact location in the world, its name, etc. This module contains the descriptor of the island data type.

## 2.10. Matrix.h

Matrix is an STL style template implementation of a two dimensional array. This construct is useful because Empire makes such heavy use of Cartesian geometry, and it is quite handy (as well as less error prone) to be able to use double subscripts, such as the sectors that make up an island.

## 2.11. Parser

The parser is a class based, recursive descent implementation of the grammar of the language used to express instructions in Empire. It reads an input text stream and emits opcodes that are placed in the queue to be executed by the game's engine.

## 2.12. Player

Player objects contain the essential data about each player: name, home island, etc.

## 2.13. Prompt

Prompt contains the console I/O functions for the game's administrative tool.

## 2.14. Scanner

A rule-based lexical scanner to be used primarily by the Parser object.

## 2.15. Sector

A sector is the atomic unit of land area in the game. Each sector “contains” the material of the land part of the game.

## 2.16. Semantics

This module contains a semantic evaluator that examines a string of parsed, syntactically correct opcodes to determine if the request is doable in the current state of the game. For example: MOVE SHIP 7 TO (233,45) is legal syntax, but it is only meaningful in a game where there is a SHIP 7, the coordinates exist, are in the water, and SHIP 7 has enough movement time to complete the task.

## 2.17. Ship

The ship object contains all the information about a single ship: type, location, heading, speed, the amount of rust, the fuel level, etc.

## **2.18.Singleton.h**

Singleton is a C++ template that allows one to create only a single copy of an object, regardless of how many copies may appear to have been made. In Empire, this allows one to repeatedly declare the Global object without worry that one has created a facsimile of the real thing.

## **2.19.Stopwatch**

A stopwatch implementation to find out how long it takes to do things.

## **2.20.Toplevel.h**

Function prototypes for the main modules that make up Empire.

## **2.21.utilities**

Templates for common operations on values.

### 3. Top Level Modules

---

#### 3.1. Close

Used to temporarily stop play within a game.

#### 3.2. Dump

The equivalent in Empire of the SQLDUMP command on a relational database.

#### 3.3. Genesis

The routine that creates a new game. Note that Open and Close still have to be used to actually make use of the output of Genesis.

#### 3.4. Main




The entry point for the game.

#### 3.5. Open

Used to cause a particular game to be available for play.

#### 3.6. Textmap

Used to create a human readable map of the game board.

Revision Number	Date of last Revision	Number of Pages
 7	 2 0 0 4 1 1 2 3 0 9 5 0	 7