**Partum Artificium Client**

Contents

[Developers 2](#_Toc272408999)

[Updating Software 2](#_Toc272409000)

[Repository 2](#_Toc272409001)

[Design 3](#_Toc272409002)

[Design Details 4](#_Toc272409003)

[PartumArtificium.Client 4](#_Toc272409004)

[PartumArtificium.Framework 4](#_Toc272409005)

[PartumArtificium.Model 4](#_Toc272409006)

[PartumArtificium.View 4](#_Toc272409007)

[PartumArtificium.Network 4](#_Toc272409008)

[Coding Standards for C# Client 5](#_Toc272409009)

[Naming Conventions 5](#_Toc272409010)

[Comments 5](#_Toc272409011)

[Third Party Tools Used in Client 6](#_Toc272409012)

[Json.NET 6](#_Toc272409013)

# Developers

Alex Brandt, Paul Segovia, Steve Longtine

# Updating Software

All updates to software must be approved before checked in. Design changes especially need to be reviewed and approved.

# Repository

<http://github.com/segoviap/Partum-Artificium-Client>

# Design

PartumArtificium.Client

PartumArtificium.Model

PartumArtificium.View

PartumArtificium

PartumArtificium.Framework

PartumArtificium.Network

# Design Details

## PartumArtificium.Client

GUI for project

## PartumArtificium.Framework

Core objects for all client code.

## PartumArtificium.Model

Core logic for GUI

## PartumArtificium.View

Should only contain interface objects. Each interface corresponds to a GUI view.

## PartumArtificium.Network

All network communication logic should be located here.

# Coding Standards for C# Client

Note 9/16/2010: This is a work in progress. New standards will be added or standards may be removed as project progresses.

## Naming Conventions

1. Use Pascal and Camel casing for naming identifiers.

|  |  |  |
| --- | --- | --- |
| Identifier | Case | Example |
| Class | Pascal | TestClass |
| Event | Pascal | FatalEvent |
| Exception | Pascal | GameException |
| Interface | Pascal | IComparable |
| Const Variable | Pascal | MaxValue |
| Private Variable | Camel | \_privateMember |
| Method | Pascal | GetOptions |
| Parameter | Camel | paramValue |
| Property | Pascal | DefaultPassword |

1. Do not abbreviate identifiers
2. Do not use casing to differentiate identifiers
3. Use noun or noun phrase to name a class or struct
   1. Optional: If class is a derived class, it is good practice to use a compound name. For example if you have class named Button, deriving from this class may result in a class named BeveledButton.
4. Preface interfaces with the letter I
5. Use a verb for naming an event
6. Use Pascal casing for naming source files

## Comments

1. Each file should have copyright information at top of document.

# Third Party Tools Used in Client

## Json.NET

<http://james.newtonking.com/projects/json-net.aspx>

<http://james.newtonking.com/projects/json/help/>

<http://www.json.org/>