# 配置

本部分专注于orx的配置系统。Orx是一个开源、跨平台、轻量、数据驱动及2D游戏引擎。

## 简介

配置系统是Orx的核心模块之一。它是使得Orx成为一个数据驱动游戏引擎的主要模块。
它是需要学习的最重要的模块之一，因为它可以在运行中调整和修改。
当你习惯它以后，你的开发时间将会大大地缩短。

它的格式和 INI 文件中的基本一致。但是添加了一些如同继承和重载的概念。 同时也添加了一些特别的操作符[1]将会在 语法部分中介绍。

## 概要

当Orx被初始化后，它会使用可执行文件的名字推断出主配置文件名。
基本上，如果有后缀名的话，它会删除可执行文件名的后缀名，然后添加 `.ini`后缀。如果你使用orx launcher (`orx.exe/orx`)，那么主配置文件名将会是orx.ini
一样的，如果你的程序名为 `MyNewGame.exe`，Orx初始化后将会加载的主配置文件名为`MyNewGame.ini`。

在主配置文件中你可以包含任何你想要的其他配置文件[2]。
你也可以直接在代码中加载配置文件。

```
orxConfig_Load("MyConfigFile.cfg");
```

一旦一个配置文件被加载，它的信息访问都会在内存中完成，文件不会再从磁盘读取除非指定调用函数`orxConfig_Load()` 或 `orxConfig_ReloadHistory()`。

你甚至可以部分或全部地保存配置的内容到一个文件。利用下面这个函数：

```
orxConfig_Save("MyConfigFile.cfg", bEncrypt, MySaveFilter);
```

`MySaveFilter`是一个在每一个配置段和关键字都会调用的回调函数，让你决定是否保存这些信息到文件。
如果你传的值为`orxNULL`而不是一个有效的回调函数，每一个单独的关键字/值都会被保存。
请注意从原来加载的文件中的原始注释不会被保存，因为在加载的时候就已经被忽略并从未存储在内存中。

在你的`MySaveFilter` 回调函数中，当提供了一个配置段的名字 和`orxNULL`作为关键字时，你要决定是只否保存这个配置段中一部分（通过返回`orxTRUE`）或者你宁愿完全地跳过整个配置段（通过返回`orxFALSE`）。
在第一种情况中，每一个配置段中的关键字/值都会被询问是否要保存。

当你想要保存部分数据，像处理保存游戏(你将在游戏保存部分中看到)等，利用这个筛选系统是非常方便的。

你可以找到doxygen格式的配置模块API通过orx's doxygen 文档页面或者直接访问这里。
我们相信API是一看便知的并且不会存在太多学习上的障碍。

配置模块的使用，数据方式（data-wise），将在下面的章节中完整地阐述。
最后一个部分将会展示如何修改它来处理保存游戏。

在许多地方配置模块都会很有用：例如它已经是本地化模块的基础。
现在由你来根据需求去发现其他巧妙的用法啦。：）

## Sections


- Syntax
- Orx's main settings
- Orx's structures settings
- Encryption & orxCrypt program
- Using config module for savegames


1)
例如，为了处理list（random或者包含文件
2)
可以是任意后缀名，`.ini` 不是强制的

From:
https://orx-project.org/wiki/ - **Orx Learning**

Permanent link:
**https://orx-project.org/wiki/cn/orx/config/main?rev=1281013175**

Last update: **2025/09/30 17:26 (14 months ago)**