Content Pack System Documentation

Piotr Paszko

2018

Contents

Chapter 1

Introduction

1.1 CPS General Description

Content Pack System (CPS) is a universal system for external resources support (i.e. game modding). System is based on "content packs" - an archive file containing resource and CPS files. CPS offers programmer API for reading content packs and loading resource files.

1.2 Content pack

Content Pack is a zip archive with *.cpack file extension. Content pack contains a pack information file, element information files and resource files.

1.2.1 Pack information file

Pack information file contains basic information about content pack formatted in XML. File have to be named exactly "Content Pack.xml". File have to be located directly inside pack archive.

Information contained:

1. Content pack name.
2. Id (formatted in domain name).
3. Version of content pack.
4. Version of CPS content pack is compatible.
5. Author of content pack.

Example file source:

```
<?xml version="1.0" encoding="utf-8"?>
<ContentPack>
  <Name>name</Name>
  <Id>name.domain.com</Id>
  <Version>v 1.0</Version>
  <CPSVersion>cpsv 1.0</CPSVersion>
```

```
  <Author>author</Author>
</ContentPack>
```

1.2.2 Element information file

Element information file contains information about a resource file with the same name. Data is formatted in XML. File extension is *.ceif. File can be located anywhere in content pack archive structure.

Information contained:

1. Resource name for displaying in application.
2. Id (formatted as subdomain).
3. Type of resource.

Example file source:

```
<?xml version="1.0" encoding="utf-8"?>
<ContentPackElement>
  <Name>name</Name>
  <Id>id</Id>
  <Type>image</Type>
</ContentPackElement>
```

1.2.3 Resource file

Resource file is a file, that is going to be used in application. Resource file have to have same name as information file describing it. Resource file also have to be at same directory as it own information file. See

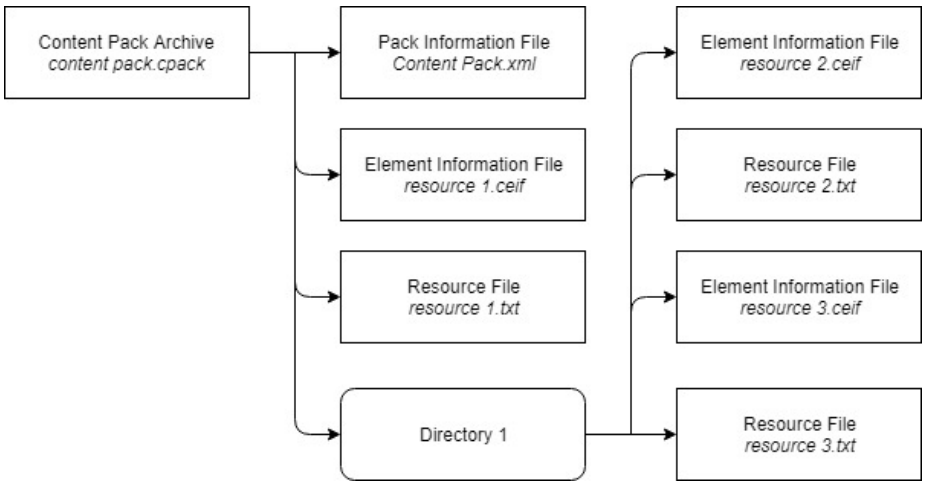

Figure 1.1: Content Pack structure example.

1.3 CPS operating principle

The cps operation is divided into three different phases:

1. Reading content packs.
2. Managing content packs.
3. Loading resources.

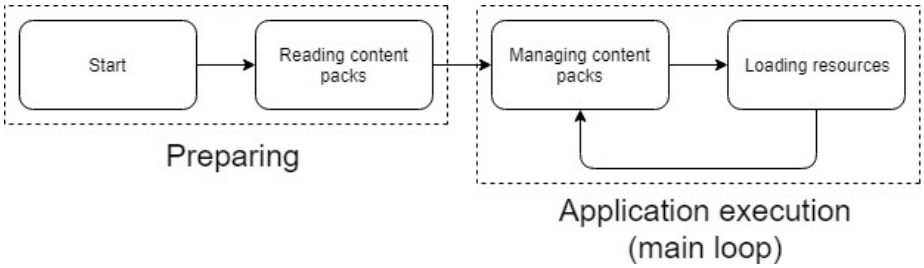

Figure 1.2: CPS operation phases.

1.3.1 Reading content packs phase

In this phase CPS reads content pack structure and all information files, and creates it own data structure presenting content pack content. In this phase programmer gets direct access to content pack objects in API. This phase should be executed in application preparing phase (for example a first loading screen in games).

1.3.2 Reading content packs phase

This phase is not performed by CPS. In this phase programmers can manage content packs and CPS reading phase warnings via API. That phase is crucial in the error handling mechanism of CPS. This phase should be executed after changes in resource packs (i.a. after reading and loading content packs phases).

1.3.3 Loading resources phase

In this phase CPS loads chosen resource and gives programmers direct access to it.

1.4 Versioning

CPS uses this versioning system:
major.minor.development stage.build

1. Major - major version number.
2. Minor - minor version number.

3. Development Stage - describes development stage of current minor version. Values meaning:
   (a) 0 - alpha,
   (b) 1 - beta,
   (c) 2 - release candidate,
   (d) 3 - release version.
4. Build - build version number.

Chapter 2

API Documentation

Need to be write after tests.

Chapter 3

Using examples

Need to be write after tests.