

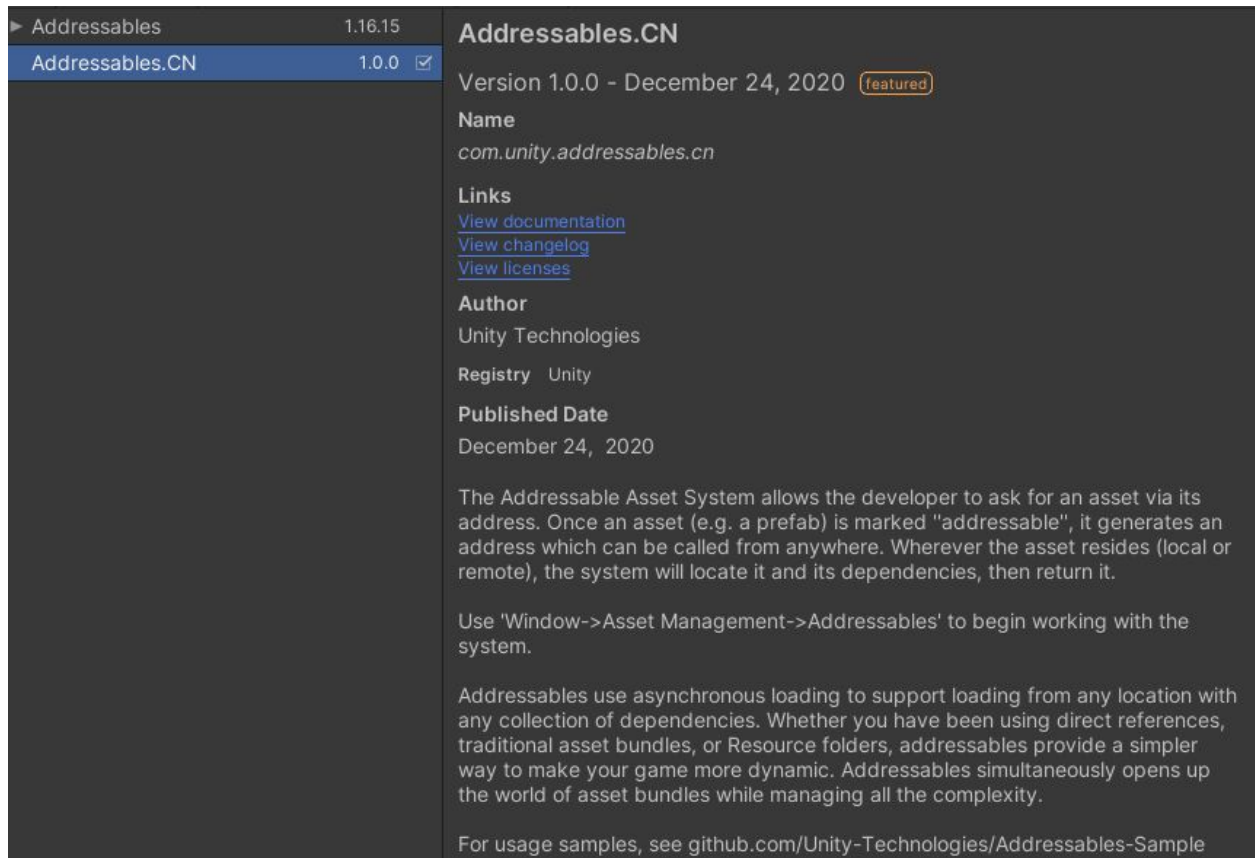
中国版可寻址资源系统的加密功能

最近我们公布了中国版可寻址资源系统。中国版本和全球版本区别就是针对部分中国用户提出的一件，增加了资源加密的功能。

这里我们介绍一下如何使用加密功能。

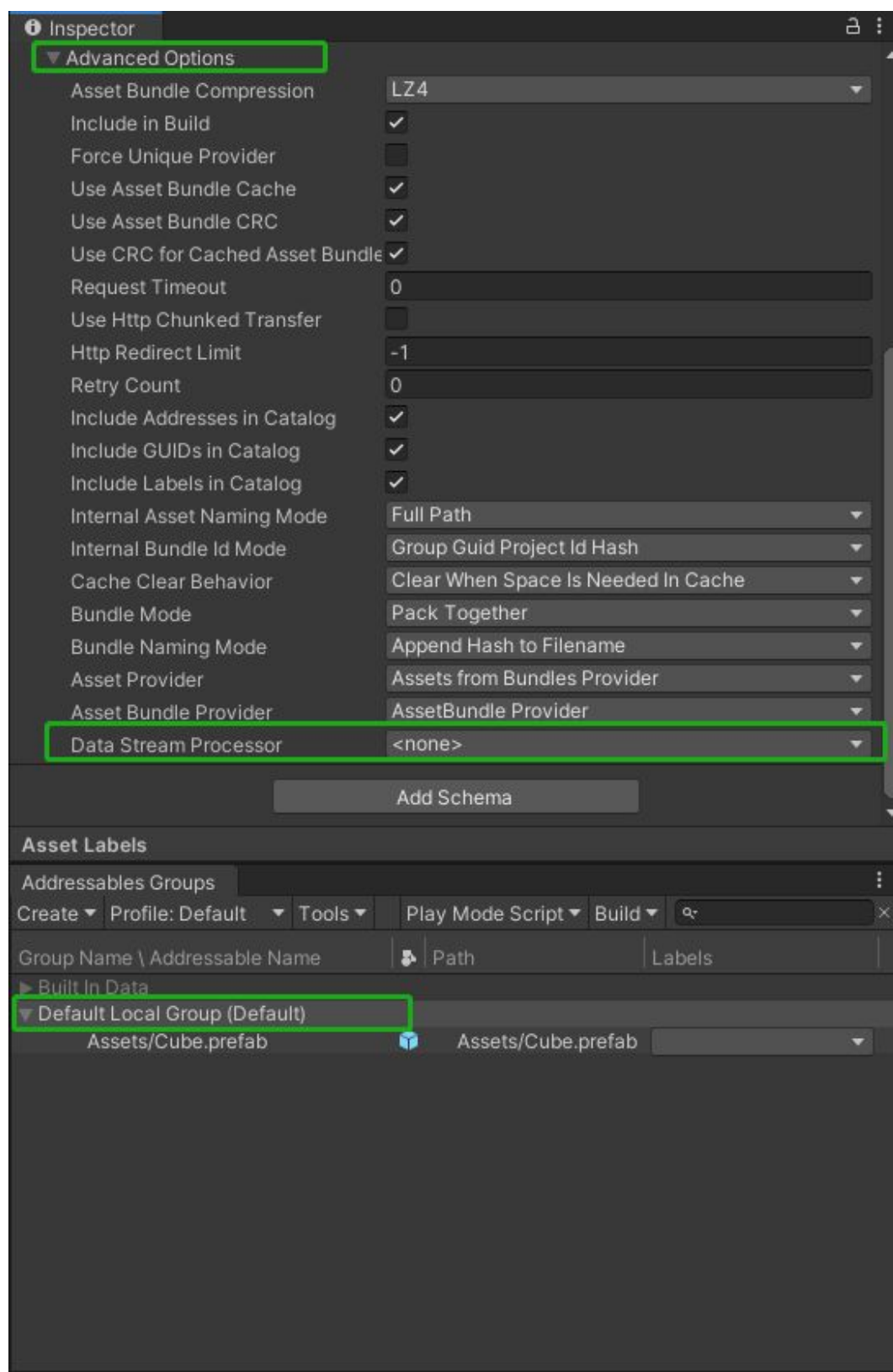
安装

首先，我们要打开Package Manager，安装Addressables.CN。



配置Local Group属性

后面的操作和国际版可寻址资源系统一样。创建Local Group。但是选择Group的高级选线会多一条属性：Data Stream Processor。

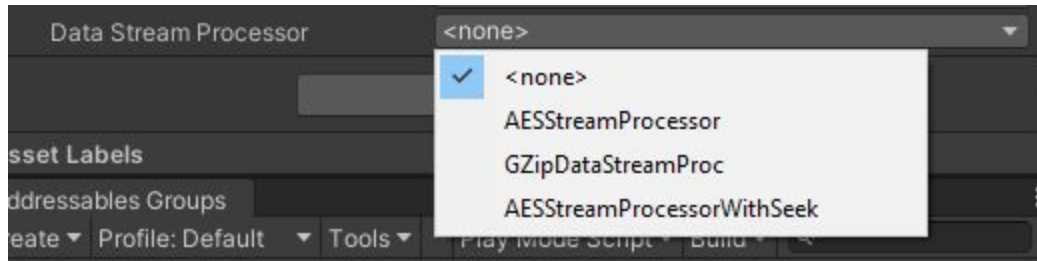


点击对应的选项，可以看到：

1. AESStreamProcessor

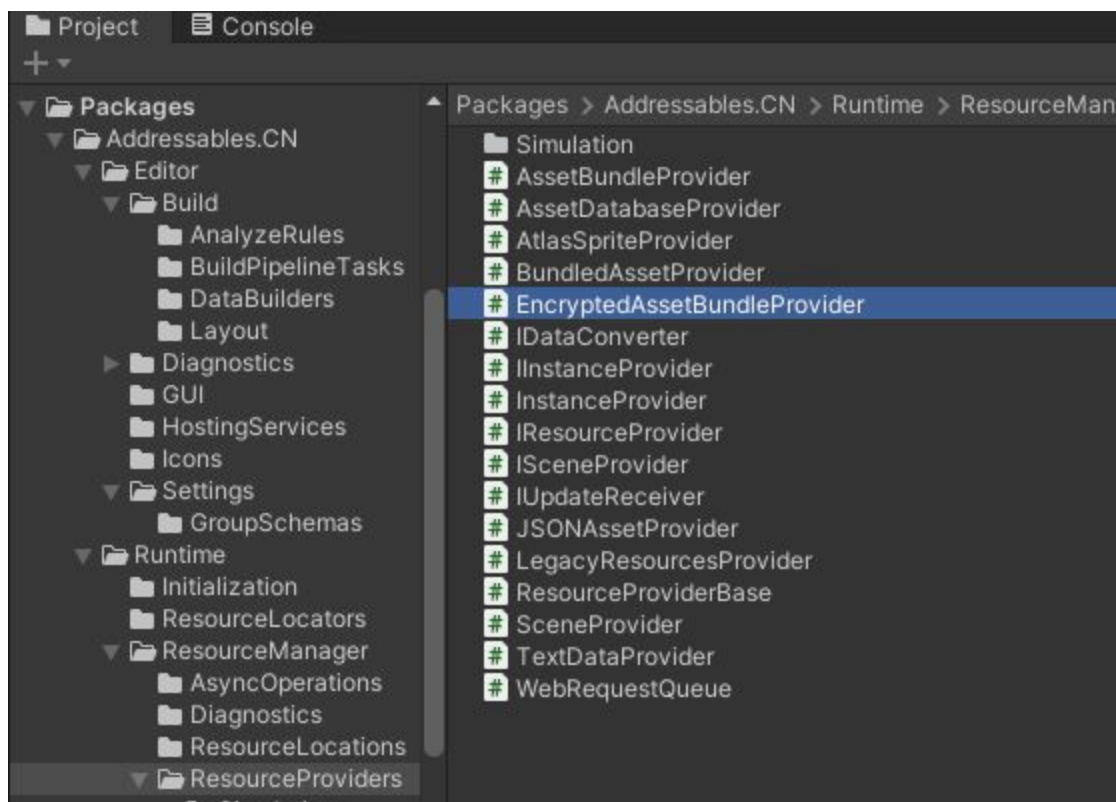
2. GZipDataStreamProc
3. AESStreamProcessorWithSeek

这三个流处理器是用来支持压缩和加密的。



自定义密码

在项目窗口中，我们可以找到他们的源代码



具体位置在项目的相对目录：

`Library\PackageCache\com.unity.addressables.cn@1.0.0\Runtime\ResourceManager\resourceProviders\EncryptedAssetBundleProvider.cs`

下面我们看一下第一个处理器：

```
public class AESStreamProcessor : IDataConverter
```

```

{
    byte[] Key { get { return
System.Text.Encoding.ASCII.GetBytes("ABCDEFGHIIJKLMNOP"); } }
    SymmetricAlgorithm m_algorithm;
    SymmetricAlgorithm Algorithm
    {
        get
        {
            if (m_algorithm == null)
            {
                m_algorithm = new AesManaged();
                m_algorithm.Padding = PaddingMode.Zeros;
                var initVector = new byte[m_algorithm.BlockSize / 8];
                for (int i = 0; i < initVector.Length; i++)
                    initVector[i] = (byte)i;
                m_algorithm.IV = initVector;
                m_algorithm.Key = Key;
                m_algorithm.Mode = CipherMode.ECB;
            }
            return m_algorithm;
        }
    }
    public Stream CreateReadStream(Stream input, string id)
    {
        return new CryptoStream(input,
Algorithm.CreateDecryptor(Algorithm.Key, Algorithm.IV),
CryptoStreamMode.Read);
    }

    public Stream CreateWriteStream(Stream input, string id)
    {
        return new CryptoStream(input,
Algorithm.CreateEncryptor(Algorithm.Key, Algorithm.IV),
CryptoStreamMode.Write);
    }
}

```

我们可以看到，逻辑很简单，就是包装了读和写的流，把数据流加密解密。我们可以看到密码是 **ABCDEFGHIIJKLMNOP** 这个密码是不能更改的，因为在已发布包中的修改会被Editor纠正。如果我们要设置自己的自定义密码，可以在项目中创建一个完全相同的类，用自己的名字，比如**MyAESSStreamProcessor**。

```
using System.IO;
using System.Security.Cryptography;

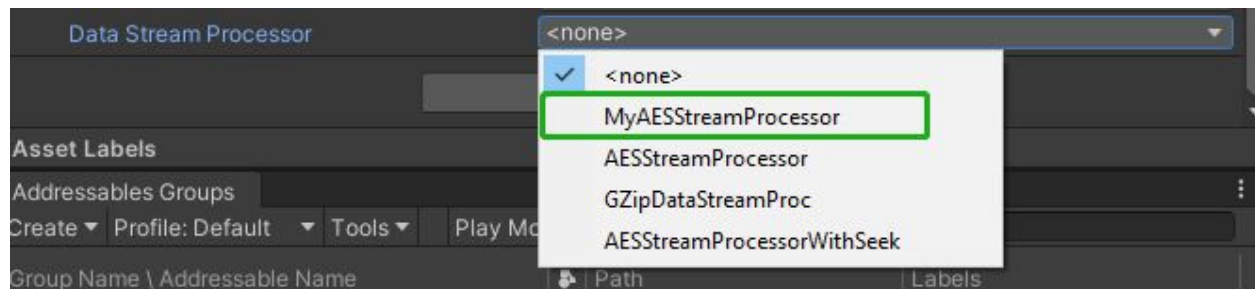
namespace UnityEngine.ResourceManagement.ResourceProviders{
    public class MyAESSStreamProcessor : IDataConverter
    {
        byte[] Key { get { return
System.Text.Encoding.ASCII.GetBytes("ABCDEFGHIIJKLMNOPXYZ"); } }
        SymmetricAlgorithm m_algorithm;
        SymmetricAlgorithm Algorithm
        {
            get
            {
                if (m_algorithm == null)
                {
                    m_algorithm = new AesManaged();
                    m_algorithm.Padding = PaddingMode.Zeros;
                    var initVector = new byte[m_algorithm.BlockSize / 8];
                    for (int i = 0; i < initVector.Length; i++)
                        initVector[i] = (byte)i;
                    m_algorithm.IV = initVector;
                    m_algorithm.Key = Key;
                    m_algorithm.Mode = CipherMode.ECB;
                }
                return m_algorithm;
            }
        }
        public Stream CreateReadStream(Stream input, string id)
        {
            return new CryptoStream(input,
Algorithm.CreateDecryptor(Algorithm.Key, Algorithm.IV),
CryptoStreamMode.Read);
        }
    }
}
```

```

        public Stream CreateWriteStream(Stream input, string id)
        {
            return new CryptoStream(input,
Algorithm.CreateEncryptor(Algorithm.Key, Algorithm.IV),
CryptoStreamMode.Write);
        }
    }
}

```

保存，刷新刚刚创建的文件（refresh），再查看Local Group的Data Stream Processor选项列表，中间就多了我们刚刚添加的处理器：



这样我们就可以用自定义密码了。

总结

利用加密处理器选项，您可以把您最宝贵的资源保护起来。在中国版可寻址资源系统中可以通过简单配置达成您的目标。

但是同时也会带来性能上的损耗。损耗主要在解密资源上面。同时加密资源内容放在 Application.persistentDataPath上，解密过程中不仅占用cpu，也会耗费额外内存空间。