**PiecePicker设计文档**

**功能及约束**

1. PiecePicker应该保存一份torrent的Piece Map，用来记录piece的下载完成状态，支持piece下载完成状态的查询。
2. PiecePicker需要按照一定算法保证所有的piece都可以被准确无误的下载完成。
3. PiecePicker必须能够确保已经被下载完的piece不应该重复下载。
4. PiecePicker需要支持接收Piece下载并校验完成后的通知。
5. PiecePicker必须确保一个block同一时刻只会被一个peer下载。
6. PiecePicker需要支持接收block下载并完成的通知，PiecePicker应该将此block标记为下载完成。
7. PiecePicker需要支持接收block下载失败的通知，PiecePicker应该将此block重新标记为未下载，保证后续还能从其他的peer处下载。

**依赖关系**

PiecePicker是一个相对独立的模块，它本身在创建后，只向外提供服务，属于被依赖的模块，具有较大的独立性。



**PiecePicker的生命周期**

PiecePicker创建的依赖数据是元数据，包括文件的总大小、piece的大小以及block的大小。但是在torrent创建的时候，元数据可能还不存在，当前我们简单处理，只有在获取了torrent的元数据后才去创建PiecePicker。Torrent创建了PiecePicker后，PiecePicker就与Torrent同生同灭了，在Torrent被销毁前，PiecePicker也会被销毁。

**PiecePicker提供的接口**（具体见源代码）

PiecePicker当前设计成所有协议共用，后续如果某些协议的策略有特殊需求，再考虑重构。

PiecePicker提供的最重要接口就是获取可以下载的block，这个接口体现了PiecePicker的最重要职能：管理和调度资源的下载过程。

**Piece/block选择策略**

当前我们按照最简单的策略实现：按顺序下载，优先下载排在前面的piece/block。后面可以在此基础上做优化。

**Block下载失败处理**

如果一个block从一个peer下载失败，则应该优先选取其他的peer去下载，如果其他的peer没有此piece，则尝试从此peer下载三次，如果三次下载都失败，则此block不再从此peer处下载。