复 用 文 档

——配置管理（CM）构件

Team 07

修改历史记录：

|  |  |  |
| --- | --- | --- |
| 编写日期 | 编写版本 | 编写作者 |
| 2015/04/20 | V1.0 | 尹巧 |
| 2015/04/30 | V2.0 | 尹巧 |
|  |  |  |

目录

**1.构件介绍3**

**2.功能说明3**

**3.使用方法4**

3.1类CM的ProcessFile方法5

3.2类CM的SearchKey方法5

3.3类CM的getSize方法5

3.4类CM的list方法5

3.5类CM的ProcessFile方法5

3.6类CM的ProcessFile方法5

3.7类CM的ProcessFile方法5

**4.注意事项4**

4.1配置文件的格式5

4.2输出结果说明5

4.3配置项重复的情况5

**1.构件介绍**

这一构件的主要作用是对MySQL的“.ini配置文件”进行配置管理。这一构件可以根据用户给定的配置文件，读取并存储配置文件中所有的配置项。并提供查询接口，用户可以利用该查询接口输入给出具体的查询结果。

**2.功能说明**

用户准备好自己想要进行配置管理的配置文件，并将该文件的路径作为参数提供给CM类的ProcessFile方法。ProcessFile方法则读取并存储该配置文件中的配置信息。（使用示例请参见下方的接口说明）

然后用户可以进行配置信息的查询，具体的操作是：以配置名作为参数提供给CM类的SearchKey方法。就可以得到查询到的配置信息。（使用示例请参见下方的接口说明）。

除了上述功能之外，CM类还提供了getSize方法（用于得到配置项的计数）和list方法（用于得到所有的配置项）。这两个方法均不需要用户提供参数。（使用示例请参见下方的接口说明）。

**3.接口说明：**

在实际的项目使用中，请先引入我们提供的CM.jar包，然后就能在程序中使用我们提供的方法，具体每个方法的使用请参加一下说明：

**3.1类CM的ProcessFile方法：**

**》作用简述：**读取用户提供的配置文件，并存储该配置文件中的所有配置项。

**》使用方法：**先实例化一个CM对象（设为cm），然后以配置文件的路径（如：“F://my-large.init”）作为cm.ProcessFile的参数，即可将从该配置文件中读取并存储参数配置。

形如：

CM cm=new CM();

cm.ProcessFile("F: \\my-large.ini");

即可。

**》注意事项：**文件的路径既可以使用绝对路径又可以使用相对路径

**3.2类CM的SearchKey方法：**

**》作用简述：**为用户提供查询配置信息的接口。

**》使用方法：**先调用CM对象的ProcessFile方法，将该配置文件的配置信息存储好，然后以用户想要查询的配置名作为cm.SearchKey参数。

形如：

CM cm=new CM();

cm.ProcessFile("F: \\my-large.ini");

cm.SearchKey("[myisamchk]write\_buffer");

即可。

**》注意事项：**用户想要查询的配置信息的输入应该包含该配置项的归属

**3.3类CM的getSize方法：**

**》作用简述：**获取配置文件中的配置项的个数。

**》使用方法：**先调用CM对象的ProcessFile方法，将该配置文件的配置信息存储好，然后调用cm.getSize()方法即可得到配置项目个数，这一方法不用提供参数。

形如：

CM cm=new CM();

cm.ProcessFile("F: \\my-large.ini");

cm.getSize();

即可。

**》注意事项：**建议应该先调用ProcessFile方法再调用getSize方法，这样才能准确得获取配置文件中的配置项，若先调用getSize方法不会报错，但是结果为0。

**3.4类CM的list方法：**

**》作用简述：**将配置文件中的所有配置项列出。

**》使用方法：**先调用CM对象的ProcessFile方法，将该配置文件的配置信息存储好，然后调用cm.list()方法即可得到配置项目个数，这一方法不用提供参数。

形如：

CM cm=new CM();

cm.ProcessFile("F: \\my-large.ini");

cm.list();

即可。

**》注意事项：**建议应该先调用ProcessFile方法再调用list方法，这样才能准确得获取配置文件中的配置项，若先调用list方法不会报错，但是结果中不会有任何配置项的展示。

**4.注意事项**

**4.1配置文件的格式**

所有的配置项均是x = y 的格式（其中x是配置名，y是配置值），并且需要保证的是每一个配置项均有一个配置归属定义，以[]声明（例如[client]）。

注：这并不意味着每一个配置项要紧跟在一个[]配置归属后面，允许一个配置归属后面跟多个连续的配置项。但是需要保证的是第一个配置项的前面是需要有一个[]配置归属定义。

配置文件中允许有注释，注释的格式为“#”打头，并且“#”为单行注释符，意为以“#”行打头的那一行全部为注释内容。

**4.2输出结果说明**

CM类的list方法中得到的所有配置项的展示是按照配置名的字典方法进行排序的，所以可能与配置项在配置文件中的排序有所不同。

**4.3配置项重复的情况**

如果有两个配置项的配置名完全一致，则越靠近文件头的配置项将会被靠近文件尾的配置项覆盖，多个配置项的配置名一致的情况也一样，最终的搜索结果和展示结果中均只会考虑最靠近文件尾的那一个配置项。