**c3p0详细配置说明** [**收藏**](JavaScript:d=document;t=d.selection?(d.selection.type!='None'?d.selection.createRange().text:''):(d.getSelection?d.getSelection():'');void(saveit=window.open('http://wz.csdn.net/storeit.aspx?t='+escape(d.title)+'&u='+escape(d.location.href)+'&c='+escape(t),'saveit','scrollbars=no,width=590,height=300,left=75,top=20,status=no,resizable=yes'));saveit.focus();)

 <c3p0-config>   
  <default-config>   
<!--当连接池中的连接耗尽的时候c3p0一次同时获取的连接数。Default: 3 -->   
<property name="acquireIncrement">3</property>   
  
<!--定义在从数据库获取新连接失败后重复尝试的次数。Default: 30 -->   
<property name="acquireRetryAttempts">30</property>   
  
<!--两次连接中间隔时间，单位毫秒。Default: 1000 -->   
<property name="acquireRetryDelay">1000</property>   
  
<!--连接关闭时默认将所有未提交的操作回滚。Default: false -->   
<property name="autoCommitOnClose">false</property>   
  
<!--c3p0将建一张名为Test的空表，并使用其自带的查询语句进行测试。如果定义了这个参数那么   
  属性preferredTestQuery将被忽略。你不能在这张Test表上进行任何操作，它将只供c3p0测试   
  使用。Default: null-->   
<property name="automaticTestTable">Test</property>   
  
<!--获取连接失败将会引起所有等待连接池来获取连接的线程抛出异常。但是数据源仍有效   
  保留，并在下次调用getConnection()的时候继续尝试获取连接。如果设为true，那么在尝试   
  获取连接失败后该数据源将申明已断开并永久关闭。Default: false-->   
<property name="breakAfterAcquireFailure">false</property>   
  
<!--当连接池用完时客户端调用getConnection()后等待获取新连接的时间，超时后将抛出   
  SQLException,如设为0则无限期等待。单位毫秒。Default: 0 -->   
<property name="checkoutTimeout">100</property>   
  
<!--通过实现ConnectionTester或QueryConnectionTester的类来测试连接。类名需制定全路径。   
  Default: com.mchange.v2.c3p0.impl.DefaultConnectionTester-->   
<property name="connectionTesterClassName"></property>   
  
<!--指定c3p0 libraries的路径，如果（通常都是这样）在本地即可获得那么无需设置，默认null即可   
  Default: null-->   
<property name="factoryClassLocation">null</property>   
  
<!--Strongly disrecommended. Setting this to true may lead to subtle and bizarre bugs.   
  （文档原文）作者强烈建议不使用的一个属性-->   
<property name="forceIgnoreUnresolvedTransactions">false</property>   
  
<!--每60秒检查所有连接池中的空闲连接。Default: 0 -->   
<property name="idleConnectionTestPeriod">60</property>   
  
<!--初始化时获取三个连接，取值应在minPoolSize与maxPoolSize之间。Default: 3 -->   
<property name="initialPoolSize">3</property>   
  
<!--最大空闲时间,60秒内未使用则连接被丢弃。若为0则永不丢弃。Default: 0 -->   
<property name="maxIdleTime">60</property>   
  
<!--连接池中保留的最大连接数。Default: 15 -->   
<property name="maxPoolSize">15</property>   
  
<!--JDBC的标准参数，用以控制数据源内加载的PreparedStatements数量。但由于预缓存的statements   
  属于单个connection而不是整个连接池。所以设置这个参数需要考虑到多方面的因素。   
  如果maxStatements与maxStatementsPerConnection均为0，则缓存被关闭。Default: 0-->   
<property name="maxStatements">100</property>   
  
<!--maxStatementsPerConnection定义了连接池内单个连接所拥有的最大缓存statements数。Default: 0  -->   
<property name="maxStatementsPerConnection"></property>   
  
<!--c3p0是异步操作的，缓慢的JDBC操作通过帮助进程完成。扩展这些操作可以有效的提升性能   
  通过多线程实现多个操作同时被执行。Default: 3-->   
<property name="numHelperThreads">3</property>   
  
<!--当用户调用getConnection()时使root用户成为去获取连接的用户。主要用于连接池连接非c3p0   
  的数据源时。Default: null-->   
<property name="overrideDefaultUser">root</property>   
  
<!--与overrideDefaultUser参数对应使用的一个参数。Default: null-->   
<property name="overrideDefaultPassword">password</property>   
  
<!--密码。Default: null-->   
<property name="password"></property>   
  
<!--定义所有连接测试都执行的测试语句。在使用连接测试的情况下这个一显著提高测试速度。注意：   
  测试的表必须在初始数据源的时候就存在。Default: null-->   
<property name="preferredTestQuery">select id from test where id=1</property>   
  
<!--用户修改系统配置参数执行前最多等待300秒。Default: 300 -->   
<property name="propertyCycle">300</property>   
  
<!--因性能消耗大请只在需要的时候使用它。如果设为true那么在每个connection提交的   
  时候都将校验其有效性。建议使用idleConnectionTestPeriod或automaticTestTable   
  等方法来提升连接测试的性能。Default: false -->   
<property name="testConnectionOnCheckout">false</property>   
  
<!--如果设为true那么在取得连接的同时将校验连接的有效性。Default: false -->   
<property name="testConnectionOnCheckin">true</property>   
  
<!--用户名。Default: null-->   
<property name="user">root</property>   
  
<!--早期的c3p0版本对JDBC接口采用动态反射代理。在早期版本用途广泛的情况下这个参数   
  允许用户恢复到动态反射代理以解决不稳定的故障。最新的非反射代理更快并且已经开始   
  广泛的被使用，所以这个参数未必有用。现在原先的动态反射与新的非反射代理同时受到   
  支持，但今后可能的版本可能不支持动态反射代理。Default: false-->   
<property name="usesTraditionalReflectiveProxies">false</property>   
  
    <property name="automaticTestTable">con\_test</property>   
    <property name="checkoutTimeout">30000</property>   
    <property name="idleConnectionTestPeriod">30</property>   
    <property name="initialPoolSize">10</property>   
    <property name="maxIdleTime">30</property>   
    <property name="maxPoolSize">25</property>   
    <property name="minPoolSize">10</property>   
    <property name="maxStatements">0</property>   
    <user-overrides user="swaldman">   
    </user-overrides>   
  </default-config>   
  <named-config name="dumbTestConfig">   
    <property name="maxStatements">200</property>   
    <user-overrides user="poop">   
      <property name="maxStatements">300</property>   
    </user-overrides>   
   </named-config>   
</c3p0-config>

在hibernate.cfg.xml文件里面加入如下的配置：  
　　<!-- 最大连接数 -->  
　　<property name="hibernate.c3p0.max\_size">20</property>  
　　<!-- 最小连接数 -->  
　　<property name="hibernate.c3p0.min\_size">5</property>  
　　<!-- 获得连接的超时时间,如果超过这个时间,会抛出异常，单位毫秒 -->  
　　<property name="hibernate.c3p0.timeout">120</property>  
　　<!-- 最大的PreparedStatement的数量 -->  
　　<property name="hibernate.c3p0.max\_statements">100</property>  
　　<!-- 每隔120秒检查连接池里的空闲连接 ，单位是秒-->  
　　<property name="hibernate.c3p0.idle\_test\_period">120</property>  
　　<!-- 当连接池里面的连接用完的时候，C3P0一下获取的新的连接数 -->  
　　<property name="hibernate.c3p0.acquire\_increment">2</property>  
　　<!-- 每次都验证连接是否可用 -->  
　　<property name="hibernate.c3p0.validate">true</property>  
　　完整示例如下（hibernate.properties）：  
　　hibernate.connection.driver\_class = org.postgresql.Driver  
　　hibernate.connection.url = jdbc:postgresql://localhost/mydatabase  
　　hibernate.connection.username = myuser  
　　hibernate.connection.password = secret  
　　hibernate.c3p0.min\_size=5  
　　hibernate.c3p0.max\_size=20  
　　hibernate.c3p0.timeout=1800  
　　hibernate.c3p0.max\_statements=50  
　　hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect

注：Hibernate除了期自身的连接池、C3PO连接池以外，还可以使用dbcp包的连接池功能（不过据说HIBERNATE3中不推荐使用，说其有BUG）、使用Proxool包的连接池功能实现。