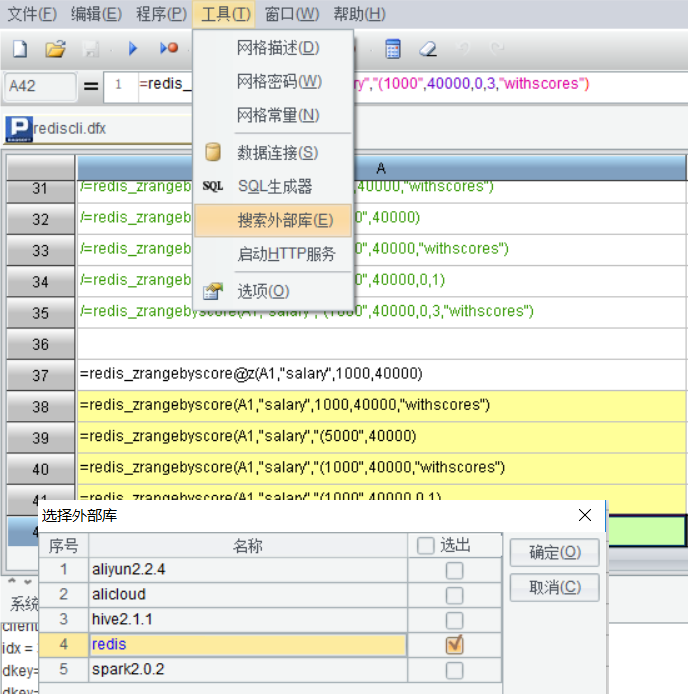
1. RedisCli与集算器接口是在redis3+StringRedisTemplate基础上开发的，接口包文件为RedisCli.3.6.1.jar。
2. 将RedisCli.3.6.1.ar包拷贝到extlib\rediscli目录下，其它的依赖包也放在此目录下。
3. 软件注册后，重启IDE软件在菜单工具选项中能看到“搜索外部库”。
4. 在IDE中搜索外部库。集算器下：
5. 

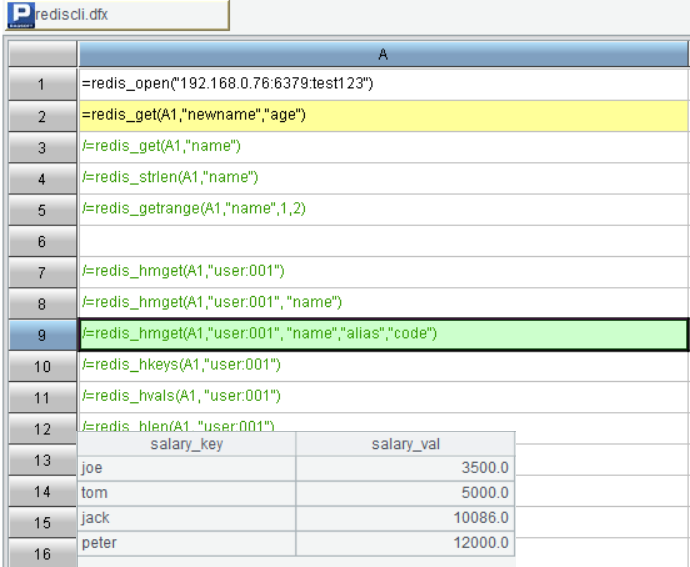
重启IDE后生效

4. jdk版本号问题

redis要求不低于jdk1.8版本支持，bin\config.txt配置参考如下：

java\_home=C:\Program Files\raqsoft\common;esproc\_port=41733;jvm\_args=-Xms128m -Xmx4866m -Duser.language=zh

5. 在集算器脚本中使用redis\_open,redis\_close,redis\_get等函数访问redis数据库，如下：



Redis\_open函数简要说明如下：

Redis\_open("spring-redis.xml")  
 redis.xml内容如下：  
 <bean id="jedisPoolConfig" class="redis.clients.jedis.JedisPoolConfig">

<property name="maxIdle" value="1" />

<property name="maxTotal" value="5" />

<property name="blockWhenExhausted" value="true" />

<property name="maxWaitMillis" value="30000" />

<property name="testOnBorrow" value="true" />

</bean>

<bean id="jedisConnectionFactory" class="org.springframework.data.redis.connection.jedis.JedisConnectionFactory">

<property name="hostName" value="192.168.0.11" />

<property name="port" value="6379"/>

<property name="poolConfig" ref="jedisPoolConfig" />

<property name="usePool" value="true"/>

<property name="password" value="rootpwd" />

</bean>

<bean id="stringRedisTemplate" class="org.springframework.data.redis.core.StringRedisTemplate">

<constructor-arg name="connectionFactory" ref="jedisConnectionFactory"/>

</bean>

**Redis连接方式：**

1. 单实例方式：  
    =redis\_open()或=redis\_open("spring-redis.xml")

B. Cluster方式：

　　=redis\_open@c()或=redis\_open("spring-cluster.xml")

C. Sentinel方式：

　 =redis\_open@s()或=redis\_open("spring-sentinel.xml")

6. redis的依赖jar包文件列表：

aopalliance-1.0.jar

commons-lang-2.1.jar

commons-logging-1.2.jar

commons-math3-3.6.1.jar

commons-pool2-2.10.0.jar

jedis-3.6.1.jar

log4j-1.2.17.jar

raq-redis-cli-3.6.jar

slf4j-api-1.7.5.jar

slf4j-log4j12-1.7.5.jar

spring-beans-5.2.0.RELEASE.jar

spring-context-5.2.0.RELEASE.jar

spring-core-5.2.0.RELEASE.jar

spring-data-commons-2.2.0.RELEASE.jar

spring-data-keyvalue-2.2.0.RELEASE.jar

spring-data-redis-2.2.0.RELEASE.jar

spring-expression-5.2.0.RELEASE.jar

spring-tx-5.2.0.RELEASE.jar

**7.　服务端启动服务：**

Redis:　redis-start &  
netstat -na|grep 6379 (缺省端口：6379)

**8.　接口定义：**1、redis\_open(xmlfile) 返回连接redis数据库句柄hd

选项s： Sentinel连接池连接方式

选项c：Cluster群集连接方式

redis\_close(hd)　关闭连接，释放资源

2、**## key**

redis\_expire(hd, key, timeout, [timeunit]) 给指定键设置过期时间  
A、timeout过期时间，long类型，缺省单位为TimeUnit.SECONDS

B、timeunit：时间单位，下面为数值与对应的时间单位  
**0:** TimeUnit.***NANOSECONDS***;

1: TimeUnit.***MICROSECONDS***;

2: TimeUnit.***MILLISECONDS***;

3: TimeUnit.***SECONDS***;

4: TimeUnit.***MINUTES***;

5: TimeUnit.***HOURS***;

6: TimeUnit.***DAYS***;

**default** :TimeUnit.***MINUTES***;  
C、当timeout date类型或date字符串（格式为yyyy-MM-dd）时，则不需要timeunit  
样例：

=redis\_expire(A1,"name",10000,2)

=redis\_expire(A1,"name","2021-07-17")

=redis\_expire(A1,"name",date("2021-07-17"))

redis\_getexpire(hd,key, [timeunit]) 获取过期时间  
样例：  
=redis\_getexpire(A1,"name")   
=redis\_getexpire(A1,"name", 4)

redis\_keys(hd, pattern) 查找所有符合给定模式pattern的key。

redis\_type(hd,key) key值的类型

redis\_persist(hd,key)　 将key持久化保存， 就是把过期或者设置了过期时间的key变为永不过期

redis\_randomkey(hd)　 随机取一个key

**## string**

redis\_get(hd,key,…) 获取一个或多个key值

redis\_getrange(hd,key, start, end)　 获取key值的子字串

样例

=redis\_getrange(A1,"name", 0, 3)

redis\_getbit(hd,key, offset)　 判断指定的offset位置ASCII码的bit位是否为1  
样例

=redis\_getbit(A1, "age", 1)

redis\_increment(hd,key,[int or double]) 给指定键 加指定整数或数值，缺省增加值为1

redis\_size(hd,key)　 获取字符串的长度

redis\_exists(hd,key)　是否存在key

**## hash**

redis\_hmget(hd,key,[field1,…])

要查询key的value值，无参数field，返回所有的字段值

若有参数field,则只返回对应的field字段的值

redis\_hkeys(hd,key)　 获取哈希表中的所有字段

redis\_hvals(hd,key)　 获取哈希表中的所有字段值

redis\_hsize(hd,key)　 获取哈希表中字段的数量

redis\_hexists(hd,key, field)　 查看哈希表中指定的字段是否存在

redis\_hscan(hd,key, regex, [count])　 迭代哈希表中的键值对

**## list**

redis\_lindex(hd,key, index)　 通过索引获取列表中的元素

redis\_lrange(hd,key, start, end)　 获取列表指定范围内的元素

redis\_ lsize (hd,key)　 获取列表长度

redis\_ltrim(hd,key) 裁剪列表指定范围内的元素

**## set**

redis\_smembers(hd, key) 获取集合所有元素

redis\_srandmember@i(hd, key, [count]) 随机获取集合中的一个元素，  
参数count：获取count个元素  
选项i: 返回去重后的元素.

redis\_sismember(hd, key, member) 查询指定键是否有该元素

redis\_sdiff(hd, key, key1 or [key1,…])key  
key集合与key1集合的差集，或key集合与多个集合的差集

redis\_sinter(hd, key, key1 or [key1,…])  
key集合与key1集合的交集，或key集合与多个集合的交集

redis\_sunion(hd, key, key1 or [key1,…])  
key集合与key1集合的并集，或key集合与多个集合的并集

redis\_sscan(hd,key, regex, [count])

获取集合的游标

**# sortSet**

redis\_zrange(hd, key, start, stop, [withscores]) @z倒序即zrevrange,   
返回有序集key中，指定区间内的成员。

redis\_zrangebyscore(hd, key, min, max, offset, n, [withscores])   
获取有序集key中，指定区间的成员。

redis\_zrank(hd, key, member) @z即zrevrank   
获取有序集key中成员member的排名。

redis\_zscore(hd, key, member)   
获取有序集key中，成员member的score值。

redis\_zcard(hd, key)

获取有序集合的成员数

redis\_zcount(hd, key, min, max)

获取集合大小

redis\_zscan(hd,key, regex, [count])

获取有序集合的游标

redis\_zsize(hd,key)

获取集合大小