# 撮合成交规则配置指引文档

# 成交规则配置方案说明

成交规则配置是指将收到订单之后需返回的消息字段配置化。

撮合系统在收到新订单后，根据匹配规则匹配到对应的配置项，取配置项中的确认、回报消息字段返回消息。匹配规则取订单消息中的任意指定字段作为匹配项目，以设定的正则表达式为匹配条件，如果所有的匹配条件均满足，那么该订单的交易使用该条配置项。配置项中的确认、回报消息字段可以引用原订单中的字段内容，也可以使用撮合系统定义的字段内容。

## 上海数据库报盘的配置

### 配置匹配条件

在规则参数页面，对需要修改的规则单击鼠标右键，在弹出的菜单中选择“配置字段”



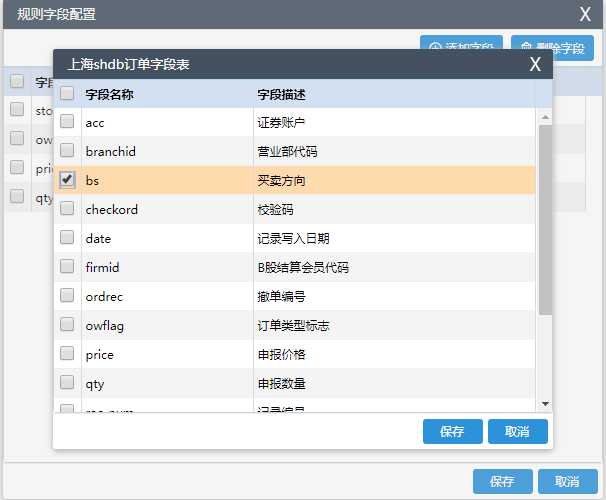


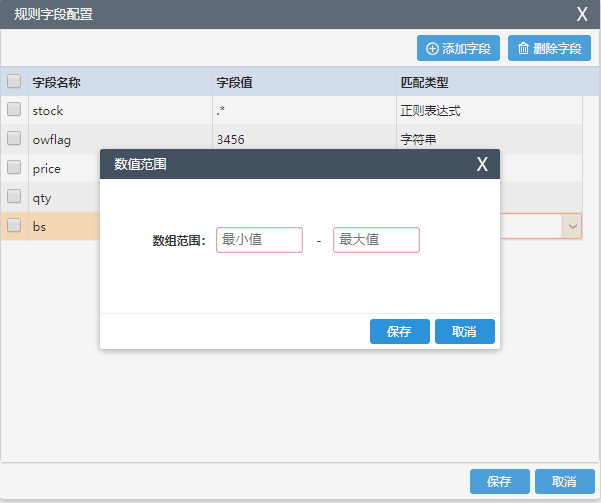
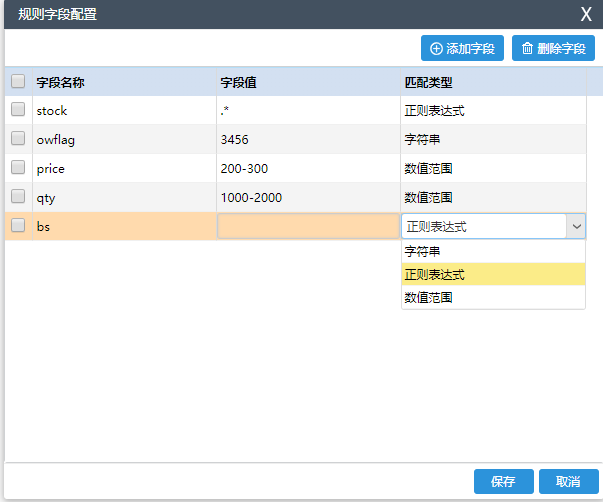
可配置的字段分为如下形式：

|  |  |
| --- | --- |
| 匹配类型 | 说明 |
| 正则表达式 | 使用“字段值”中填写的正则表达式对“字段名称”所标注的字段进行匹配 |
| 字符串 | 使用“字段值”中填写的字符串对“字段名称”所标注的字段进行匹配 |
| 数值范围 | 使用“字段值”中的数值范围对“字段名称”所标注的字段进行匹配 |

相关操作：

（1）添加、修改字段：点击“添加字段”后会先让用户选择所需判断的字段名称，勾选后点击“保存”，然后直接修改字段值即可。如需切换“匹配类型”，直接下拉“匹配类型”列表修改；如果选择的数值范围，会在弹出中输入最小值和最大值。





（2）删除字段：勾选需要删除的字段，点击“删除字段”。

### 规则匹配说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 规则ID | 订单字段 | 匹配类型 | 字段值 | 说明 |
| rule\_1 | stock | 数值范围 | 600400-600499 | 在600400-600499之间的数字 |
| owflag | 字符串 | LPT | “LPT”的字符串 |
| price | 正则表达式 | .\* | 匹配除了回车符和换行符之外的所有字符任意多次 |
| qty | 数值范围 | 200-300 | 在200-300之间的数字 |
| rule\_2 | stock | 数值范围 | 710000-710999 | 在710000-710999之间的数字 |
| owflag | 字符串 | WTH | “WTH”的字符串 |
| price | 正则表达式 | .\* | 匹配除了回车符和换行符之外的所有字符任意多次 |
| qty | 正则表达式 | .\* | 匹配除了回车符和换行符之外的所有字符任意多次 |

\*注：正则表达式请参考文后附录1.

按以上配置:

1. 新订单stock=600431，owflag=LPT，任意price、qty的订单将按规则rule\_1返回确认、回报消息.
2. 新订单stock=710233，owflag=WTH，任意price、qty的订单将按规则rule\_2返回确认、回报消息。

### 配置返回消息字段



返回消息分为确认回报和成交回报，分别对应上海数据库报盘的【申报确认接口 ordwth2】和【成交回报接口 cjhb】中的字段。**每条成交回报配置对应一个回报，成交回报可以添加多条，或者也可以无回报（对应无回报的场景）**。

撮合系统按回报规则中填写的内容，经过撮合系统转译，写入对应的接口的对应字段，如：在成交回报中【gsdm(席位号)】的内容填写为“123456”，在回报的接口中【gsdm】就写入“123456”。

回报规则中填写内容支持的定义如下：

（1）需要引用申报订单的数据时，以@ +所需字段引用，如reff, acc, stock, bs, price, qty, status, owflag, ordrec, firmid, branchid等申报表对应数据。

（2）需要系统生成变量，如订单合同号、订单交易时间等，以$+预设变量引用：

|  |  |
| --- | --- |
| 系统预设变量 | 系统预设变量说明 |
| DATE | 日期，格式为YYYYMMDD |
| REC\_NUM | 记录编号 |
| SH\_CJBH | 成交编号 |
| TIME\_C6 | 时间，格式为HHMMSS |
| TIME\_C8 | 时间，格式为HH:MM:SS |

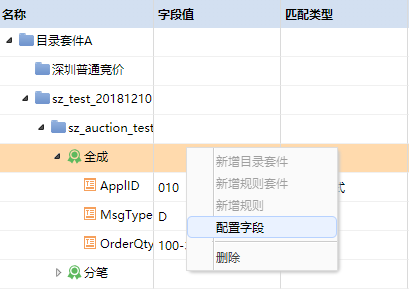
（3）任意的字符串，系统将直接原样写入回报中。

（4）在表达式内部允许使用四则运算，仅支持+-×/，**各个字段之间要用空格做分割**。如要得到成交的总金额，可用： @price \* @qty；要得到一半的成交数量，可用：@qty / 2。

## 深圳EzSTEP的配置

### 1.2.1. 配置匹配条件

在规则参数页面，对需要修改的规则单击鼠标右键，在弹出的菜单中选择“配置字段”



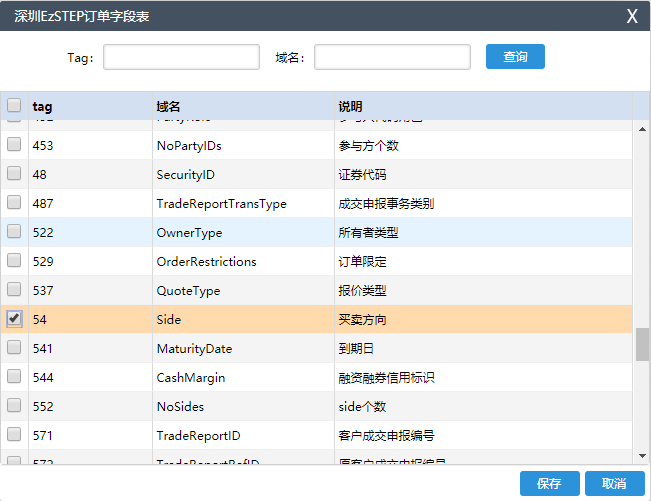


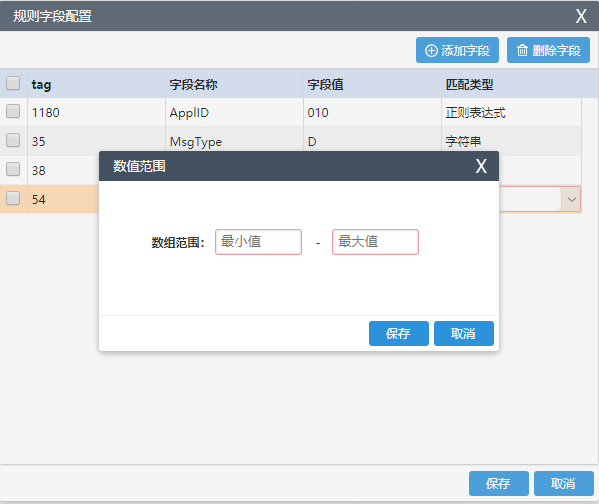
可配置的字段分为如下形式：

|  |  |
| --- | --- |
| 匹配类型 | 说明 |
| 正则表达式 | 使用“字段值”中填写的正则表达式对“字段名称”所标注的字段进行匹配 |
| 字符串 | 使用“字段值”中填写的字符串对“字段名称”所标注的字段进行匹配 |
| 数值范围 | 使用“字段值”中的数值范围对“字段名称”所标注的字段进行匹配 |

相关操作：

（1）添加、修改字段：点击“添加字段”后会先让用户选择所需判断的字段名称，勾选后点击“保存”，然后直接修改字段值即可。如需切换“匹配类型”，直接下拉“匹配类型”列表修改；如果选择的数值范围，会在弹出中输入最小值和最大值。

（2）删除字段：勾选需要删除的字段，点击“删除字段”。

### 1.2.2. 规则匹配说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 规则ID | 订单字段 | 匹配类型 | 字段值 | 说明 |
| rule\_1 | ApplID | 字符串 | 010 | “010”的字符串 |
| MsgType | 字符串 | D | “D”的字符串 |
| OrderQty | 正则表达式 | .\* | 匹配除了回车符和换行符之外的所有字符任意多次 |
| rule\_2 | ApplID | 字符串 | 120 | “120”的字符串 |
| MsgType | 字符串 | D | “D”的字符串 |
| OrderQty | 正则表达式 | .\* | 匹配除了回车符和换行符之外的所有字符任意多次 |

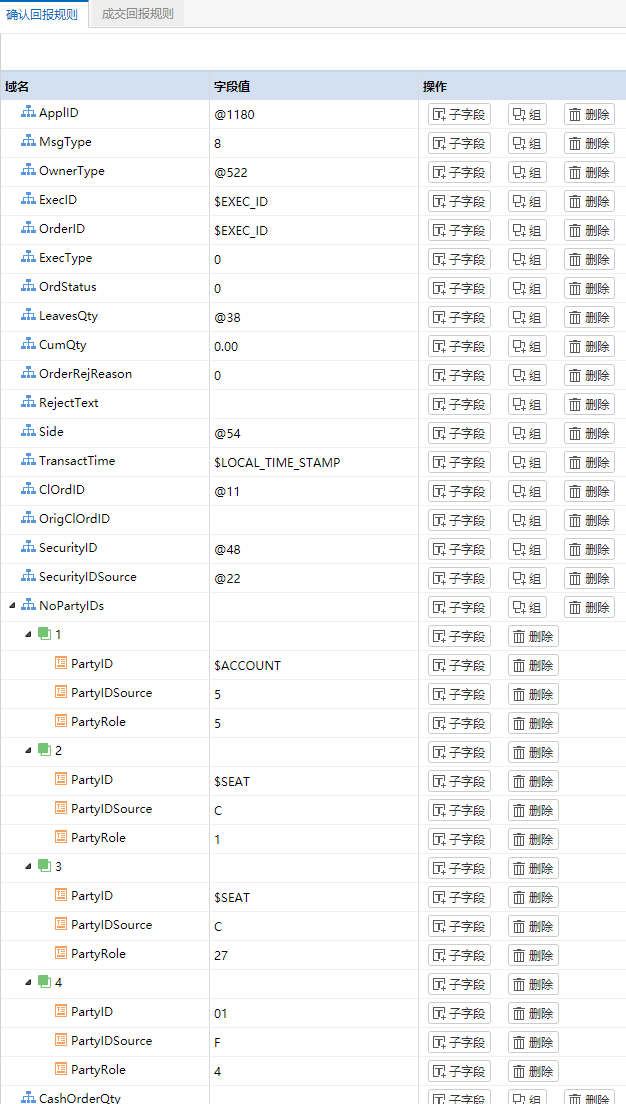
\*注：正则表达式请参考文后附录1.

按以上配置:

（1）新订单ApplID=010，MsgType=D，任意OrderQty的订单将按规则rule\_1返回确认、回报消息。

（2）新订单ApplID=120，MsgType=D，任意OrderQty的订单将按规则rule\_1返回确认、回报消息。

### 配置返回消息字段



返回消息分为确认回报和成交回报，按STEP交易数据的规则构造返回字段。**每条成交回报配置对应一个回报，成交回报可以添加多条，或者也可以无回报（对应无回报的场景）**。

撮合系统按回报规则中填写的内容，经过撮合系统转译，写入对应的接口的对应字段，如：在成交回报中【ApplID】的内容填写为“010”，在回报的消息中【1180】字段就写入“010”。

回报规则中填写内容支持的定义如下：

（1）需要引用申报订单的数据时，以@ +所需字段的Tag值引用，如44(Price),54(Side)等申报表对应数据。

（2）需要系统生成变量，如订单合同号、订单交易时间等，以$+预设变量引用：

|  |  |
| --- | --- |
| 系统预设变量 | 系统预设变量说明 |
| ACCOUNT | 交易账户(此变量由原订单重复组中取得) |
| SEAT | 交易席位(此变量由原订单重复组中取得) |
| EXEC\_ID | ExecID C16 交易所赋予的执行编号，单个交易日内不重复 |
| LOCAL\_TIME\_STAMP | LocalTimeStamp C21 本地时间戳 YYYYMMDD-HH:MM:SS.sss(毫秒) |
| ORDER\_ID | OrderID C16 交易所赋予的订单编号，跨交易日不重复 |

（3）任意的字符串，系统将直接原样写入回报中。

（4）在表达式内部允许使用四则运算，仅支持+-×/，**各个字段之间要用空格做分割**。如要得到成交的总金额，可用： @44(price)\* @38(OrderQty)；要得到一半的成交数量，可用：@38(OrderQty) / 2。

# 附录1：正则表达式在线测试网址

<https://c.runoob.com/front-end/854>

# 附录2：正则表达式配置说明

正则表达式(regular expression)描述了一种字符串匹配的模式（pattern），可以用来检查一个串是否含有某种子串、将匹配的子串替换或者从某个串中取出符合某个条件的子串等。

例如：

* **runoo+b**，可以匹配 runoob、runooob、runoooooob 等，+ 号代表前面的字符必须至少出现一次（1次或多次）。
* **runoo\*b**，可以匹配 runob、runoob、runoooooob 等，\* 号代表字符可以不出现，也可以出现一次或者多次（0次、或1次、或多次）。
* **colou?r** 可以匹配 color 或者 colour，? 问号代表前面的字符最多只可以出现一次（0次、或1次）。

构造正则表达式的方法和创建数学表达式的方法一样。也就是用多种元字符与运算符可以将小的表达式结合在一起来创建更大的表达式。正则表达式的组件可以是单个的字符、字符集合、字符范围、字符间的选择或者所有这些组件的任意组合。

正则表达式是由普通字符（例如字符 a 到 z）以及特殊字符（称为"元字符"）组成的文字模式。模式描述在搜索文本时要匹配的一个或多个字符串。正则表达式作为一个模板，将某个字符模式与所搜索的字符串进行匹配。

**普通字符**

普通字符包括没有显式指定为元字符的所有可打印和不可打印字符。这包括所有大写和小写字母、所有数字、所有标点符号和一些其他符号。

**特殊字符**

所谓特殊字符，就是一些有特殊含义的字符，如上面说的 runoo\*b 中的 \*，简单的说就是表示任何字符串的意思。如果要查找字符串中的 \* 符号，则需要对 \* 进行转义，即在其前加一个 \: runo\\*ob 匹配 runo\*ob。

许多元字符要求在试图匹配它们时特别对待。若要匹配这些特殊字符，必须首先使字符"转义"，即，将反斜杠字符\ 放在它们前面。下表列出了正则表达式中的特殊字符：

|  |  |
| --- | --- |
| 字符 | 描述 |
| $ | 匹配输入字符串的结尾位置。如果设置了 RegExp 对象的 Multiline 属性，则 $ 也匹配 '\n' 或 '\r'。要匹配 $ 字符本身，请使用 \$ |
| ( ) | 标记一个子表达式的开始和结束位置。子表达式可以获取供以后使用。要匹配这些字符，请使用 \( 和 \) |
| \* | 匹配前面的子表达式零次或多次。要匹配 \* 字符，请使用 \\* |
| + | 匹配前面的子表达式一次或多次。要匹配 + 字符，请使用 \+ |
| . | 匹配除换行符 \n 之外的任何单字符。要匹配 . ，请使用 \. |
| [ | 标记一个中括号表达式的开始。要匹配 [，请使用 \[ |
| ? | 匹配前面的子表达式零次或一次，或指明一个非贪婪限定符。要匹配 ? 字符，请使用 \? |
| \ | 将下一个字符标记为或特殊字符、或原义字符、或向后引用、或八进制转义符。例如， 'n' 匹配字符 'n'。'\n' 匹配换行符。序列 '\\' 匹配 "\"，而 '\(' 则匹配 "(" |
| ^ | 匹配输入字符串的开始位置，除非在方括号表达式中使用，此时它表示不接受该字符集合。要匹配 ^ 字符本身，请使用 \^ |
| { | 标记限定符表达式的开始。要匹配 {，请使用 \{ |
| | | 指明两项之间的一个选择。要匹配 |，请使用 \| |

**限定符**

限定符用来指定正则表达式的一个给定组件必须要出现多少次才能满足匹配。有 \* 或 + 或 ? 或 {n} 或 {n,} 或 {n,m} 共6种。

正则表达式的限定符有：

|  |  |
| --- | --- |
| 字符 | 描述 |
| \* | 匹配前面的子表达式零次或多次。例如，zo\* 能匹配 "z" 以及 "zoo"。\* 等价于{0,} |
| + | 匹配前面的子表达式一次或多次。例如，'zo+' 能匹配 "zo" 以及 "zoo"，但不能匹配 "z"。+ 等价于 {1,} |
| ? | 匹配前面的子表达式零次或一次。例如，"do(es)?" 可以匹配 "do" 、 "does" 中的 "does" 、 "doxy" 中的 "do" 。? 等价于 {0,1} |
| {n} | n 是一个非负整数。匹配确定的 n 次。例如，'o{2}' 不能匹配 "Bob" 中的 'o'，但是能匹配 "food" 中的两个 o |
| {n,} | n 是一个非负整数。至少匹配n 次。例如，'o{2,}' 不能匹配 "Bob" 中的 'o'，但能匹配 "foooood" 中的所有 o。'o{1,}' 等价于 'o+'。'o{0,}' 则等价于 'o\*' |
| {n,m} | m 和 n 均为非负整数，其中n <= m。最少匹配 n 次且最多匹配 m 次。例如，"o{1,3}" 将匹配 "fooooood" 中的前三个 o。'o{0,1}' 等价于 'o?'。请注意在逗号和两个数之间不能有空格 |

示例：

由于章节编号在大的输入文档中会很可能超过九，所以您需要一种方式来处理两位或三位章节编号。限定符给您这种能力。下面的正则表达式匹配编号为任何位数的章节标题：

|  |
| --- |
| /Chapter [1-9][0-9]\*/ |

请注意，限定符出现在范围表达式之后。因此，它应用于整个范围表达式，在本例中，只指定从 0 到 9 的数字（包括 0 和 9）。

这里不使用 + 限定符，因为在第二个位置或后面的位置不一定需要有一个数字。也不使用 ? 字符，因为使用 ? 会将章节编号限制到只有两位数。您需要至少匹配 Chapter 和空格字符后面的一个数字。

如果您知道章节编号被限制为只有 99 章，可以使用下面的表达式来至少指定一位但至多两位数字。

|  |
| --- |
| /Chapter [0-9]{1,2}/ |

上面的表达式的缺点是，大于 99 的章节编号仍只匹配开头两位数字。另一个缺点是 Chapter 0 也将匹配。只匹配两位数字的更好的表达式如下：

|  |
| --- |
| /Chapter [1-9][0-9]?/ |

或

|  |
| --- |
| /Chapter [1-9][0-9]{0,1}/ |

\*、+限定符都是贪婪的，因为它们会尽可能多的匹配文字，只有在它们的后面加上一个?就可以实现非贪婪或最小匹配。

例如，您可能搜索 HTML 文档，以查找括在 H1 标记内的章节标题。该文本在您的文档中如下：

|  |
| --- |
| <H1>Chapter 1 - 介绍正则表达式</H1> |

**贪婪**：下面的表达式匹配从开始小于符号 (<) 到关闭 H1 标记的大于符号 (>) 之间的所有内容。

|  |
| --- |
| /<.\*>/ |

**非贪婪**：如果您只需要匹配开始和结束 H1 标签，下面的非贪婪表达式只匹配 <H1>。

|  |
| --- |
| /<.\*?>/ |

如果只想匹配开始的 H1 标签，表达式则是：

|  |
| --- |
| /<\w+?>/ |

通过在 \*、+ 或 ? 限定符之后放置 ?，该表达式从"贪心"表达式转换为"非贪心"表达式或者最小匹配。