
昆明方元利科技有限责任公司

<MisExpress 业务解释平台>

配置文件编写说明文档

版本 <1.0>

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

修订历史记录

日期	版本	说明	作者
<月/日/年>	<x.x>	<详细信息>	<姓名>
6/5/2003	<1.0>		谢道多

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

配置文件编写说明文档

1. 说明

- ✧ 此文档只针对 XML 配置文件的编写做说明
- ✧ 使用本文档时在 Word 中请选择使用文档结构图，在 AcrobatReader 中请使用标签
- ✧ XML 由元素组成.一个元素包括标签和属性.如字符串“<name>Mary</name>”称为 name 元素,其中 name 为标签，Mary 为属性
- ✧ 所有标签和属性都是大小写敏感的，这是由 XML 本身决定的
- ✧ 一个标签如果还没有很好的被支持，以颜色显示
- ✧ 一个标签如果比较重要，以黑体字显示
- ✧ 所有标签符合 Java 命名规则，首字母小写，以后每个单词的第一个字母大写。属性的书写规则由配置文件里定义。

2. 约定

- ✧ 配置文件中严格区分大小写
- ✧ Enumerate 类型在配置文件中用字符串表示
- ✧ Boolean 类型使用字符串“true”和“false”表示
- ✧ 在配置文件中的所有名称不能相同
- ✧ 关键字：

关键字	使用位置	描述
key	field.format	表示这个 field 实例的值
ThisRowChange	relations_Linkage.initiative	表示网格的当前行发生了变化
Procedure	saveStruct.objectName	执行存储过程的调用（有返回值的调用）
sum	relations_Linkage.functionString	汇总
GetIndexString	relations_Linkage.functionString	形如： value := GetIndexString (“1:[str1];2:[str2]”, 1) 取得了“str1”
LoopGetIndexString	relations_Linkage.functionString	形如： value := GetIndexString (“1:[str1];2:[str2]”) 循环取值
GetDetailString	relations_Linkage.functionString	形如：value := GetDetailString (“str1,str2”, 1) 取得了“str1” 或 value := GetDetailString (“str1#str2”, @, 1) 取得了“str1” “@”为分隔符号，它可以是一个字符
GetPartString	relations_Linkage.functionString	形如： value := GetPartString (“1:[str11, str12];2:[str21, str22]”, 1, 2) 取得了“str12”
Default	sign.value	表示状态的默认定义（取得的 value 没有被定义时使用）
actionCommander	Commander.name	单独定义的一个按钮，它在传递参数后执行
initCommander	Commander.name	单独定义的一个按钮，它在初始化对象后执行
<comma>	application	应用间的参数传递时使用，用于解析,”

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

✧ 流程顺序

1) 页面启动后：

第一步：执行为对象付初始值、

第二步：执行 initCommander 定义的动作（如果有 initCommander 的定义）。

第三步：传递参数（为变量付值）、

第四步：执行 actionCommander 定义的动作（如果有 actionCommander 的定义）。

2) 点击按钮时：

第一步：提示用户（questionMessage）

第二步：变量付值（relations_Linkage）

第三步：检查操作（showMessages）

第四步：相关应用调用（relaApp）

第五步：存储结构（saveStruct）

第六步：业务操作（operateAction）

第七步：状态位处理（signExpression）

第八步：抛出参数（popParameter）

第九步：对话操作（talkAction）

3) 保存结构：先执行存储过程调用，然后顺序执行查询语句

4) 业务操作：

第一步：changeSignSQL

第三步：affectSQL

第四步：logSQL

第五步：refAction

第六步：workflow

✧ 在页面中不能有名为“SubmitForm”的 Form

✧ 工作流调用约定

从任务列表中取出数据时，系统会自动向变量付值：变量为“DocID”,“TOrder”

3. XML 基础

在这里编写 XML 配置文件不需要对 XML 了解很多。XML 是一种可扩展的标记语言。与 HTML 相比，XML 的元素是完全可以自定义的。

注意 XML 文件的必须是良好格式的文档。

例如：

<input type="text" name="text1">

在 HTML 中是合法的，但是在 XML 中就不是，因为 input 标签没有关闭。应该这样：

<input type="text" name="text1"/>

注意在配置文件的书写中一定要将打开的标签关闭。

有一些特殊的字符应该按以下方式进型特殊字符的转义：

字符	转义字符
<	<
>	>
&	&
'	'

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

“	"
---	--------

例如：

```
...
<exp>a>b</exp>
```

...
是非法的。正确的写法如下：

```
...
<exp>a&gt;b</exp>
```

...
以上这个方式很难阅读，可以将包含大量的特殊字符的内容包含在“<![CDATA[”和“]]>”之间，表示其中的内容不需要解析。

```
<exp>
<![CDATA[
a>b>c
]]>
</exp>
```

4. XML 配置文件结构

application

这是每个配置文件的根元素且无属性。

子元素：

元素名	类型	必要	多个	说明	默认值
name	String	是	否	应用的名称	无
htmlFile	String	是	否	对应的网页文件名	无
valueClass	valueClass	否	是	变量定义	\
oneLineEdit	oneLineEdit	否	是	单行编辑器，是网页上的一组有关联的编辑对象的容器	\
textMatrix	TextMatrix	否	是	网格定义	\
commander	Commander	否	是	按钮定义	\
noteOperatedState	NoteOperatedState	否	否	状态位定义	\
keyDefine	KeyDefine	否	否	键盘定义	\

Name

应用的名称，与 XML 文件的名称一致。

htmlFile

XML 文件对应的 HTML 文件，一般为“XML 文件名称.htm”。

valueClass

应用的变量定义

<Mi sExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	Edi tDate: <6/21/2003>

子元素：

元素名	类型	必要	多个	说明	默认值
Name	String	是	否	变量名称， 建议参数名以 v_开头	无
fieldType	Enumerate	是	否	数据类型 fieldTypeEnumerate	无

系统内建的全局变量名只有 1 个

变量名	类型	描述	说明
UserID	String	登录用户的 ID 号	系统内建，用户登录后产生

可向表 fyl_PublicValues 里加入其它自定义的全局变量。但是所定义的全局变量必须由全局变量 **UserID** 引申出来。譬如增加全局变量 StaffName 相应的 SQL 语句如下：

```
INSERT INTO fyl_PublicValues (ProjectID,ValueName,ValueType,QuerySQL,FillOrder)
VALUES ('zmProject','StaffName','String','select user_name from zm_zd_user where
user_code=<@>%UserID%<@>,1)
```

注意：在不同的系统中 SQL 语句会相应的不同。同时要注意 FillOrder，譬如如果定义了全局变量 CorpCode,而全局变量 CorpName 由 CorpCode 作为查询条件得到，那么 FillOrder 决定的顺序必须先生成全局变量 CorpCode,然后才能生成 CorpName.

特殊变量有以下 3 个，其名称为关键字，不能在自定义变量中取以下 3 个名称！

(a) v_NoteFlag：代表单据某一状态

(b) DocID：公文编号。新建时,它会在内部获得一个值 (DocID := %Client_code%)

流转时则会对外部付值 (Client_code := %DocID%)，所以要定义两个

relations_Linkage

(c) Torder：作为流转时从外部获得的工作序号

例：<!--单据状态-->

```
<valueClass>
  <name> v_NoteFlag </name>
  <fieldType> String </fieldType>
</valueClass>
```

附：

<!--从 DocID 中获得单位编号、客户编号-->

```
<relations_Linkage>
  <initiative>DocID</initiative>
  <linkageType>FormatControl</linkageType>
  <functionString>Client_code := GetDetailString ("%DocID%", 2)</functionString>
</relations_Linkage>
```

```
<relations_Linkage>
  <initiative>DocID</initiative>
  <linkageType>FormatControl</linkageType>
  <functionString>Corp_code := GetDetailString ("%DocID%", 1)</functionString>
</relations_Linkage>
```

<!--对 DocID 付值-->

```
<relations_Linkage>
  <initiative>Client_code</initiative>
  <initiative>Corp_code</initiative>
  <linkageType>StringPlus</linkageType>
  <functionString>DocID := %Corp_code% %Client_code%</functionString>
```

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

</relations_Linkage>

name

变量名称。建议变量名以 v_ 开头,v 代表 valueClass。

变量名称不能取系统内建全局变量(3 个)、自定义全局变量(n 个)和特殊变量(3 个)！

fieldType

变量类型。支持的变量类型如下：

值	描述
Long	整型
double	浮点型
boolean	布尔型
String	字符串型
datetime	日期型

oneLineEdit

单行编辑器，是网页上的一组有关联的编辑对象的容器

子元素：

元素名	类型	必要	多个	说明	默认值
Name	String	是	否	单行编辑器名称	无
keyValueName	String	否	是	定义单行编辑器的主键，可以是 ValueClass 或 Field 的名称	无
field	Field	是	是	对应网页中的编辑框	\
relations_Linkage	Relations Linkage	否	是	复合关系及多应用之间的关系处理	\
updateSQL	String	否	否	存储数据时使用的 update 语句	无
insertSQL	String	否	否	存储数据时使用的 insert 语句	无
DeleteSQL	String	否	否	存储数据时使用的 delete 语句	无
CreateSQL	String	否	否	创建新记录时并且取得一个键值时使用	无

name

单行编辑器名称。可随意定义。此名称会被配置文件其它地方引用。建议 oneLineEdit_ 开头

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

keyValueName

定义单行编辑器的主键，可以是 ValueClass 或 Field 的名称。

field

编辑域

子元素：

子元素	类型	必要	多个	说明	默认值
Name	String	是	否	名称（必须对应于网页上的编辑框）	无
InitValue	String	否	否	初值（可以是全局变量或常量）	无
fieldType	Enumerate	是	否	数据类型 fieldType Enumerate	无
editType	Enumerate	是	否	编辑类型 editType Enumerate	无
format	String	否	否	格式化字符串、也可以在这写查询语句查询显示值。 关键字“%key%”表示操作值。	无
maxLength	Long	否	否	文本编辑框中的最大字长	无
readOnly	Boolean	否	否	初始化时的编辑状态 fieldType 为 Text 和 Textarea 才会有此属性。 如果其它 FieldType 使用了此属性那么此属性当作 disabled 属性处理(兼容处理)。	无
disabled	Boolean	否	否	初始化时的编辑状态。 所有的 FieldType 包括 Text 和 Textarea 都支持此属性	无
fillListSQL	String	否	否	填充下拉框的查询语句，只能用全局变量作为查询条件	无
fillItem	String	否	是	下拉框中的默认填充项，格式：“操作值,显示值”	无
nextField	String	否	否	焦点控制时使用	无

编辑域包括以下几中类型：(编辑域的类型由 editType 定义)

值	描述
---	----

<Mi sExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	Edi tDate: <6/21/2003>

Text	同 html 定义
Password	同 html 定义
Select	同 html 定义
Textarea	同 html 定义
Checkbox	同 html 定义
Radio	同 html 定义
Hidden	同 html 定义
Date	日期型，在网页中使用 Text

%key%这个系统内建变量用来代表当前 field 的值(看下面的示例)

除了 Hidden 类型，其它的类型必须同时在 HTML 文件中也有定义。譬如此处定义一个文本框，那么 HTML 文件上必须有一个同样名字的文本框与它对应。但是如果是 Hidden 类型，则不需要在 HTML 文件中定义。

例如：

```
<field>
  <name>Corp_code</name>
  <initValue>%CorpCode%</initValue>
  <fieldType>String</fieldType>
  <editType>Text</editType>
  <format>select corp_name from zm_zy_corp where corp_code = '%key%'</format>
  <readOnly>true</readOnly>
</field>
```

日期形的 field 的格式如下：（注意大小写）

```
<field>
  <name>Apply_date</name>
  <fieldType>datetime</fieldType>
  <editType>Date</editType>
  <readOnly>>false</readOnly>
</field>
```

name

编辑域名称。名称与 HTML 文件中的编辑域的名称一致。

initValue

初始值（可以是全局变量或常量）

全局变量：<initValue>%CorpName%</initValue>

或者

<initValue>1</initValue>

fieldType

数据类型。与 valueClass 的数据类型一样：

值	描述
long	长整型
double	浮点型
boolean	布尔型
String	字符串型
datetime	日期型

fieldType 说明：

(1) **情况 1(Text 类型)**：该情况的特点是：Text 的内容来自于数据库，且显示值和保存值不一

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

样

形如：<format>select Corp_name from zm_zy_corp where corp_code = '%key%'</format>

- (2) **情况 2(Text 类型)**：该情况的特点是：数据库中得到的 field 的值(%key%)反映到页面上时的转换，使之具有可识别性。

形如：<format>select case %key% when '1' then '稽查员' when '2' then '户管员'
end</format>

- (3) **情况 3(Select 类型)**：该情况的特点是：以下拉框的形式显示数据，且显示值与保存值不一样。

- a. 固定值：配置示例：

```
<fillItem>--请选择--</fillItem>
<fillItem>1,科员</fillItem>
<fillItem>2,科长</fillItem>
```

- b. 来自于数据库：配置示例：

```
<fillListSQL>select Region_code, Region_name from zm_zd_region where corp_code =
'%CorpCode%' </fillListSQL>
<fillItem>--请选择--</fillItem>
```

在查询语句中，<fillListSQL>结果集输出格式为 select value, caption from tablename where ...

editType

编辑域类型。

值	描述
Text	同 html 定义
Password	同 html 定义
Select	同 html 定义
Textarea	同 html 定义
Checkbox	同 html 定义
Radio	同 html 定义
Hidden	同 html 定义
Date	日期型，在网页中使用 Text

format

格式化字符串、也可以在这写查询语句查询显示值。

例如：

```
<format>select case %key% when '1' then '烟机'
when '2' then '烟叶' when '3' then '烟草制品' when '4' then '辅料' when '5' then '卷烟' end
```

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

</format>

maxLength

当类型为 text 时使用，定义文本编辑框中的最大字长

<maxLength>20</maxLength>

readOnly

初始化时的编辑状态，定义页面刚打开的时候当前编辑域是否允许被修改。只能为 true 或者 false. fieldType 为 Text 和 Textarea 才会有此属性.如果其它 FieldType 使用了此属性那么此属性当作 disabled 属性处理(兼容处理).

<readOnly>>false</readOnly>

disabled

初始化时的编辑状态，定义页面刚打开的时候当前编辑域是否可用。只能为 true 或者 false

<disabled>>false</disabled>

fillListSQL

填充下拉框的查询语句。因为初始化的时候执行此 SQL，没有自定义变量可用，所以**只能用全局变量作为查询条件**。

在查询语句中，<fillListSQL>结果集输出格式为 select value, caption from tablename where ...

<fillListSQL>

```
select corp_code, corp_name from zm_zy_corp where super_corp_code =
'%CorpCode%' or corp_code = '%CorpCode%'
```

</fillListSQL>

fillItem

下拉框中的默认填充项，格式：“操作值,显示值”

<fillItem>--请选择--</fillItem>

<fillItem>01,一般纳税人</fillItem>

<fillItem>02,小规模纳税人</fillItem>

nextField

焦点控制时使用。按 Tab 键后下一个焦点所在的 field 名称。**一般不需要使用，默认即可。**

<nextField>checkNo</nextField>

relations_Linkage

复合应用。复合关系及多应用之间的关系处理

子元素：

子元素	类型	必要	多个	说明	默认值
initiative	String	否	是	Field 和 ValueClass 实例的名称或关键字 “ThisRow_Change”	无
linkageType	Enumerate	是	否	关联类型 linkageType Enumerate	无

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

functionString	String	是	否	合法的运算定义	无
filiale	String	否	否	只处理 LoopGetIndexString 是使用	无
restrictObject	String	否	否	被限制的对象（编辑类型为 Select 的 field.name）	无
commanderName	String	否	否	按钮调用	无

```

<relations_Linkage>
  <initiative>Client_code</initiative>
  <initiative>Corp_code</initiative>
  <linkageType>StringPlus</linkageType>
  <functionString>DocID := %Corp_code%@%Client_code%</functionString>
</relations_Linkage>

```

initiative

field 或 valueClass 实例的名称或关键字“ThisRow_Change”。当 field 或者 valueClass 值或者网格中焦点行发生改变的时候，开始触发这个处理，如果值未发生改变则不触发。

linkageType

关联类型：

值	描述
Query	<p>查询，主要用于查询结果集，并填充 field 的值。包括填充单行编辑器和网格中的 field</p> <pre> <relations_Linkage> <initiative>Corp_code</initiative> <initiative>Client_code</initiative> <linkageType>Query</linkageType> <functionString>select apply_person as Apply_person, stuff(stuff(convert(char(10), apply_date, 102), 5, 1, '-'), 8, 1, '-') as Apply_date, busyness_no as Busyness_no, corporation as Corporation, full_name as Full_name, abbr_name as Abbr_name, address as Address, id_card as Id_card, reg_money as Reg_money, tax_no as Tax_no, bank as Bank, account as Account, postcode as Postcode, tel as Tel, sale_dep as Sale_dep, inspector_code as Inspector_code, corp_format as Corp_format, business_style as Business_style, region_corp_code as Region_corp_code, region as Region, </pre>

<Mi sExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	Edi tDate: <6/21/2003>

	<pre> business_range as Business_range, tax_pay_code as Tax_pay_code, trade as Trade, istown as Istown, city as City, county as County, town as Town, village as Village, person_num as Person_num, licence_no as Licence_no from dps_zy_client where corp_code = '%Corp_code%' and client_code = '%Client_code%'</functionString> </relations_Linkage> </pre>
Restrict	<p>限制一个下拉框的填充数据，与其对应的 functionString 是一个同 Field.fillListSQL 相同的查询语句（结果集结构相同）</p> <p>譬如由下拉框“片区所属单位”发生改变触发下拉框“当前单位所属的片区”发生相应改变的配置如下</p> <pre> <relations_Linkage> <initiative>Region_corp_code</initiative> <linkageType>Restrict</linkageType> <functionString>select region_code, region_name from zm_zd_region where corp_code = '%Region_corp_code%'</functionString> <restrictObject>Region</restrictObject> </relations_Linkage> </pre>
Caculate	<p>计算，主要用于相关 field 或 valueClass 的计算关联</p> <pre> <relations_Linkage> <initiative>sum1</initiative> <linkageType>Caculate</linkageType> <functionString>v_sum1 := sum(%sum1%)</functionString> </relations_Linkage> </pre>
StringPlus	<p>字符串操作，主要用于相关 field 或 valueClass 的字符串相加，形如：</p> <p>DocID := % Corp_code%@%Client_code%</p>
FormatControl	<p>格式化字符串处理，他与 GetIndexString，LoopGetIndexString，GetDetailString，GetPartString 配合使用</p> <pre> <relations_Linkage> <initiative>DocID</initiative> <linkageType>FormatControl</linkageType> <functionString>Client_code := GetDetailString ("%DocID%", @, 2)</functionString> </relations_Linkage> </pre>

linkageType 说明：

(1) **StringPlus**: 字符串的付值。示例：

```

<relations_Linkage>
<initiative>corp_code</initiative>
<initiative>begin_time</initiative>
<initiative>end_time</initiative>
<initiative>note_flag</initiative>
<linkageType>StringPlus</linkageType>

```

<Mi sExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	Edi tDate: <6/21/2003>

```

<functionString>v_str1 :=<![CDATA[ where corp_code = '%corp_code%'
and stuff(stuff(convert(char(10),in_date, 102), 5, 1,
'-'), 8, 1, '-') >= '%begin_time%'
and
stuff(stuff(convert(char(10),in_date, 102), 5, 1, '-'), 8, 1, '-') <= '%end_time%'
and flag like '%note_flag%'
]]>
</functionString>

```

</relations_Linkage>

红色标记的字符是 xml 中的防解析符，主要对 ' ' , ' ' 的处理；

用这种类型的关系处理，可以满足对变量值的传递，同时可以把多个关系字符付值给一个变量（如把查询条件付给一个变量），这样做可以减少不必要的关联动作

(2) **Restrict** : 表示其限制了一个下拉框的填充数据，functionString 是一个同 Field.fillListSQL 相同的查询语句（结果集结构相同）示例：

<relations_Linkage>

<initiative>Region_corp_code</initiative> //片区所属单位 限制对象

<linkageType>Restrict</linkageType>

<functionString>select Region_code, region_name from zm_zd_region where corp_code = '%Region_corp_code%'</functionString>

<restrictObject>Region</restrictObject> //所属片区 被限制对象

</relations_Linkage>

(3) **Query** : 当 field 的值来自于数据库时，用该类型，但只限于 oneLineEdit 或 textMatrix 的内部。示例：

<relations_Linkage>

<initiative>Client_code</initiative>

<linkageType>Query</linkageType>

<functionString>select a.apply_person as Apply_person , a.apply_date as Apply_date, ... from dps_zy_client a where corp_code = '%CorpCode%' and Client_code = '%Client_code%'</functionString>

</relations_Linkage>

functionString

合法的运算定义

譬如：

```

<functionString>select region_code, region_name from zm_zd_region where corp_code = '%Region_corp_code%'</functionString>

```

```

<functionString>v_sum1 := sum(%sum1%)</functionString>

```

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

<functionString>Client_code := GetDetailString ("%DocID%", @, 2)</functionString>

restrictObject

被限制的对象（编辑类型为 Select 的 field.name）

```
<relations_Linkage>
  <initiative>Region_corp_code</initiative>
  <linkageType>Restrict</linkageType>
  <functionString>select region_code, region_name from zm_zd_region where corp_code
= '%Region_corp_code%'</functionString>
  <restrictObject>Region</restrictObject>
</relations_Linkage>
```

filiale

只处理 LoopGetIndexString 是使用

commanderName

按钮调用

<commanderName>已定义的按钮名称</commanderName>

updateSQL

存储数据时使用的 update 语句

```
<updateSQL>update dps_zy_client set apply_person = '%Apply_person%',
      apply_date = '%Apply_date%',
      busyness_no = '%Busyness_no%',
      corporation = '%Corporation%',
      full_name = '%Full_name%',
      abbr_name = '%Abbr_name%',
      address = '%Address%',
      id_card = '%Id_card%',
      reg_money = '%Reg_money%',
      tax_no = '%Tax_no%',
      bank = '%Bank%',
      account = '%Account%',
      postcode = '%Postcode%',
      tel = '%Tel%',
      business_style = '%Business_style%',
      business_range = '%Business_range%',
      tax_pay_code = '%Tax_pay_code%',
      trade = '%Trade%',
      istown = '%Istown%',
      city = '%City%',
      county = '%County%',
      town = '%Town%',
      village = '%Village%',
      person_num = '%Person_num%',
      sale_dep = '%Sale_dep%',
      inspector_code = '%Inspector_code%',
```

<Mi sExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	Edi tDate: <6/21/2003>

```

            inspector_name = '%Inspector_name%',
            licence_no = '%Licence_no%'
where corp_code = '%Corp_code%'
and client_code = '%Client_code%'</updateSQL>

```

insertSQL

存储数据时使用的 insert 语句

```

<insertSQL>insert dps_zy_client (corp_code,
                                client_id,
                                client_code,
                                apply_person,
                                apply_date,
                                busyness_no,
                                corporation,
                                full_name,
                                abbr_name,
                                address,
                                id_card,
                                reg_money,
                                tax_no,
                                bank,
                                account,
                                postcode,
                                tel,
                                corp_format,
                                business_style,
                                region_corp_code,
                                region,
                                business_range,
                                tax_pay_code,
                                trade,
                                istown,
                                city,
                                county,
                                town,
                                village,
                                person_num,
                                sale_dep,
                                inspector_code,
                                inspector_name,
                                licence_no,
                                flag)
values ('%Corp_code%',
        '%v_Client_id%',
        '%Client_code%',
        '%Apply_person%',
        '%Apply_date%',
        '%Busyness_no%',
        '%Corporation%',
        '%Full_name%',
        '%Abbr_name%',

```


<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

```
'%Address%',
'%Id_card%',
'%Reg_money%',
'%Tax_no%',
'%Bank%',
'%Account%',
'%Postcode%',
'%Tel%',
'%Corp_format%',
'%Business_style%',
'%Region_corp_code%',
'%Region%',
'%Business_range%',
'%Tax_pay_code%',
'%Trade%',
'%Istown%',
'%City%',
'%County%',
'%Town%',
'%Village%',
'%Person_num%',
'%Sale_dep%',
'%Inspector_code%',
'%Inspector_name%',
'%Licence_no%',
'0')
</insertSQL>
```

deleteSQL

存储数据时使用的 delete 语句

```
<deleteSQL>
delete from dps_zy_client where corp_code = '%Corp_code%' and client_code =
'%Client_code%'
</deleteSQL>
```

createSQL

创建新记录时并且取得一个键值时使用

```
<createSQL>select %CorpCode% as corp_code, 10 as v_NoteFlag</createSQL>
```

showMessage

提示信息，在用户点击按钮时被调用

子元素：

子元素	类型	必要	多个	说明	默认值
name	String	是	否	名称	无
objectName	String	否	否	field.name, 定义被检查的field	无

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

condition	Enumerate	否	否	检查的类别: Not Null, Unique, Not Exist, Logic Check	无
exp	String	否	否	逻辑检查时使用的表达式	无
tableName	String	否	否	数据库中的表名 (当 condition != Not Null 时使用)	无
fieldName	String	否	是	表中的字段名	无
message	String	否	否	检查不通过时显示给用户的信息	无

```

<showMessage>
  <name>sn_Created_date</name>
  <objectName>Created_date</objectName>
  <condition>Not Null</condition>
  <message>发放日期，不能为空！</message>
</showMessage>

```

name

名称。自定义。一个应用里必须唯一。会被引用。showMessage_corpCode

objectName

field.name, 定义被检查的 field

condition

检查的类别:

值	描述
Not Null	不能为空
Unique	数据库中唯一
Not Exist	存在检查 (数据库中必须存在)
Logic Check	逻辑检查

Not Null:

```

<showMessage>
  <name>sn_Created_date</name>
  <objectName>Created_date</objectName>
  <condition>Not Null</condition>
  <message>发放日期，不能为空！</message>
</showMessage>

```

Unique:

```

<showMessage>
  <name>su_Client_code</name>
  <objectName>Client_code</objectName>
  <condition>Unique</condition>

```

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

```
<tableName>dps_zy_client</tableName>
<fieldName>corp_code = '%CorpCode%'</fieldName>
<fieldName>client_code = '%Client_code%'</fieldName>
<message>客户编号值有重复,请重新输入!</message>
</showMessage>
```

Not Exist:

暂无

Logic Check:

exp

逻辑表达式。当 condition 为 Logic Check 时使用。譬如结束日期必须大于开始日期，则表达式为（ endDate 和 beginDate 分别为两个 field. ）：

```
<exp>endDate > beginDate</exp>
<exp>%sum1% + %sum3%> %sum2%</exp>
```

当以上逻辑为假时，将 message 的消息发送给客户端，并取消客户端的请求。

tableName

数据库中的表名（当 condition != Unique 时使用）

fieldName

表中的字段名

message

检查不通过时显示给用户的信息

textMatrix

网格。对应于 html 中的 table.

子元素：

元素名	类型	必要	多个	说明	默认值
name	String	是	否	名称，要与网页中的对应网格名称相同	无
gridStyle	Enumerate	是	否	网格样式类型： gridStyle Enumerate	无
keyValueName	String	是	否	主键，可以是 ValueClass 或 Field 的名称	无
foreignKeyValueName	String	是	否	外键，可以是 ValueClass 或 Field 的名称	无
field	Field	是	是	对应网页中的编辑框	\
relations_Linkage	Relations_Linkage	否	是	复合关系及多应用之间的关系处理	\
zeroNotSave	String	否	否	field.name 当该字段为 0	无

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

				时不保存	
fillSQL	String	是	否	填充网格的查询语句	无
updateSQL	String	否	否	存储数据时使用的 update 语句	无
insertSQL	String	否	否	存储数据时使用的 insert 语句	无
deleteSQL	String	否	否	存储数据时使用的 delete 语句	无
parameterStyle	String	否	是	格式：value.name := string;	无
showMessage	ShowMessage	否	是	提示信息，在用户点击按钮时被调用	\
maxRowCount	Long	否	否	指定网格在客户端最大行数（分页使用）	无

name

网格名称。必须与 html 中定义的 table 的 id 的名称一致。

gridStyle

网格类型：

值	描述
none	普通的网格
radio	可以选择单行的网格
check	可选择多行的网格

gridStyle 说明：

gridStyle	updateSQL	insertSQL	deleteSQL
none	执行所有（过滤为零的行）	执行所有（过滤为零的行）	删除所有（无过滤）
radio	执行被选中的（无过滤）	执行被选中的（无过滤）	执行被选中的（无过滤）
check	执行被选中的（过滤为零的行）	执行被选中的（过滤为零的行）	执行被选中的（无过滤）

当 **gridStyle = radio** 时必须有一个名称为 TextMatrix.Name + "_Radio" 的 Field。其类型为 radio，如果需要对当前行进行关系处理时则需要在 relations_Linkage 的 initiative 中定义为 "ThisRow_Change"，示例：

```
<relations_Linkage>
  <initiative>ThisRow_Change</initiative>
  <linkageType>StringPlus</linkageType>
  <functionString>v_str3 := where v.goods_code = d.goods_code and
d.corp_code = '%g_corp_code%' and d.in_no = '%g_in_no%'</functionString>
```

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

</relations_Linkage>

注：在网格中尽量避免在 field 中配置 format，可以在网格的填充语句（fillSQL）中完成各 field 的填充。可以使用 case 语句或多表关联即可。

keyValueName

主键，可以是 ValueClass 或 Field 的名称

foreignKeyValueNames

外键，可以是 ValueClass 或 Field 的名称

field

同 onLineEdit 中的 field

relations_Linkage

同 onLineEdit 中的 relations_Linkage

zeroNotSave

网格中某 field 的 name，当该 field 值为 0 时不保存当前记录

```
<zeroNotSave> g_wgsl</zeroNotSave>
<zeroNotSave> g_wpdj</zeroNotSave>
```

fillSQL

填充网格的查询语句

```
<fillSQL>select goods_type as g_goodsType,
      goods_no as g_goodsNo,
      goods_name as g_goodsName,
      cigarette_type as g_cigaretteType,
      amount as g_Amount,
      unit as g_Unit,
      price as g_Price,
      money as g_Money,
      check_amount as g_checkAmount,
      case_dispose as g_caseDispose,
      cigarette_disposal as g_cigaretteDisposal
from zm_gl_case_goods_new
where corp_code = '%param_corpcode%'
and case_id = '%param_caseid%'
</fillSQL>
```

updateSQL

存储数据时使用的 update 语句

insertSQL

存储数据时使用的 insert 语句

deleteSQL

存储数据时使用的 delete 语句

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

parameterStyle

格式：value.name := string;

showMessage

建议参数名以“sn_”开头

同 onLineEdit 中的 showMessage

maxRowCount

指定网格在客户端最大行数（分页使用）

commander

按钮。特殊按钮：

子元素：

子元素	类型	必要	多个	说明	默认值
name	String	是	否	名称	无
enabled	Boolean	是	否		无
refresh	Boolean	否	否	为真时重新刷新网页	无
questionMessage	String	否	否	提示用户确认的信息，如：真的要删除该记录吗？请确认！	无
showMessages	String	否	否	对 oneLineEdit 和 textMatrix 中的 showMessage 的调用，它实际上是检查的调用。 格式： “1:[object.name,showMessage.name];.....”	无
relations_Linkage	Relations_Linkage	否	是	复合关系及多应用之间的关系处理，但按钮被点击时才作此关系处理	\
relaApp	RelaApp	否	否	相关应用调用定义	\
saveStruct	SaveStruct	否	是	存储结构定义	\
operateAction	OperateAction	否	是	业务操作	\
signExpression	String	否	否	状态表达式 select flag as v_Flag from ... where ... 返回结果集	无
succeedMessage	String	否	否	执行成功提示，如：新增成功！	无
lostMessage	String	否	否	执行失败提示，如：新增失败！	无
popParameter	String	否	是	抛出参数定义，向创建者发射变量的改变,在被调用的应用中定义 格式：1:[output,	无

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

				%v_valueClassName%]	
talkAction	TalkAction	否	是	对话操作定义	/
objectControl	String	否	是	对象控制	无

name

按钮名称。除特殊按钮外，名称必须与 HTML 中按钮的名称一致。

enabled

页面刚打开时此按钮是否可用。

refresh

为 true 时重刷新网页。效果上页面会重新初始化并出现闪动现象

questionMessage

提示用户确认的信息，如：真的要删除该记录吗？请确认！

showMessages

对 oneLineEdit 和 textMatrix 中的 showMessage 的调用，它实际上是检查的调用。

格式：“1:[object.name,showMessage.name];……”

<showMessages>

1:[oneLineEdit_Issue, sn_Created_date];2:[oneLineEdit_Issue, sn_Valid_date]

</showMessages>

relations_Linkage

同 onLineEdit 中的 relations_Linkage

relaApp

相关应用调用定义

子元素：

子元素	类型	必要	多个	说明	默认值
appName	String	是	否	应用名称	无
className	String	是	否	类型名（注：使用配置文件的页面都是“DataPond”）	无
top	Long	否	否	顶边距	无
left	Long	否	否	左边距	无
height	Long	否	否	高	无
width	Long	否	否	宽	无
printFlag	Boolean	是	否	是否是打印页面	无
windowType	Enumerate	是	否	窗口样式: windowType Enumerate	无
messages	String	否	否	消息，如：“1:[参数名,	无

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

				值]...”，建议参数名以 param_开头	
--	--	--	--	---------------------------	--

```
<relaApp>
  <appName>zm_client_address</appName>
  <className>DataPond</className>
    <top>200</top>
    <left>100</left>
    <height>900</height>
    <width>850</width>
    <printFlag>>false</printFlag>
    <messages>1:[param_address,%v_Address%]</messages>
</relaApp>
```

appName

应用名称

className

类型名（注：使用配置文件的页面都是“DataPond”）

top

顶边距

left

左边距

height

高

width

宽

printFlag

是否是打印页面

windowType

值	描述
None	普通类型，需要读取 top, left, height, width
FillScreen	全屏幕窗口，不需定义 top, left, height, width
Center	居中窗口，需要读取 height, width
noPOP	直接的本页提交，不是弹出窗口

Messages

消息，如：“1:[参数名,值]...”，建议参数名以 param_ 开头

<messages>1:[param_address,%v_Address%],2:</messages>

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

saveStruct

存储结构定义

子元素：

子元素	类型	必要	多个	说明	默认值
objectName	String	是	否	是一个 TextMatrix or OneLineEdit 实例的名称 或是 “Procedure”表示存储过程的调用	无
saveSQL	Enumerate、String	是	否	保存类型： saveSQL Enumerate 当 objectName = Procedure 是 那么 SaveSQL 描述存储过程的调用格式为： 1:[valueClassName];... := ProcedureName 1:[valueClassName1];...	无

objectName

是一个 TextMatrix or OneLineEdit 的名称 或是 “Procedure”表示存储过程的调用

示例：当 objectName = ‘Procedure’时，在 saveSQL 的等号右边是存储过程的传入参数，左边是输出参数。

```
<saveStruct>
  <objectName>Procedure</objectName>
  <saveSQL>1:[v_Client_id];2:[v_Client_code];3:[v_Licence_no] :=
p_zm_get_clientid 1:[CorpCode];2:[v_Sale_dep];3:[v_Corp_format]</saveSQL>
</saveStruct>
```

saveSQL

保存类型：[saveSQL Enumerate](#)

当 objectName = Procedure 是 那么 SaveSQL 描述存储过程的调用格式为：

1:[valueClassName];... := ProcedureName 1:[valueClassName1];...

类型：

UpdateSQL	调用 object.updateSQL
InsertSQL	调用 object.insertSQL
DeleteSQL	调用 object.deleteSQL

以上无 oneLineEdit 中的 createSQL

operateAction

业务操作

子元素：

子元素	类型	必要	多个	说明	默认值
changeSignSQL	String	否	否	修改状态 SQL，不返回结果集	无
affectSQL	String	否	否	相关影响操作 SQL，不返回结果集	无

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

logSQL	String	否	否	事务记录 SQL，不返回结果集	无
refAction	String	否	否	其它相关的操作.用于调用事务服务（保留）	无
workflow	Workflow	否	否	工作流调用	无

```
<operateAction>
  <changeSignSQL>update dps_zy_client set flag = '5' where corp_code =
'%v_Corp_code%' and client_code = '%v_Client_code%'
</changeSignSQL>
  <workflow>
    <workflowCard>1:[ProjectID,%ProjectID%];2:[CorpCode,%v_Corp_code%];3:[DocID,%DocID %];4:[StaffID,%UserID%];5:[DocType,许可
证];6:[GroupID,2];7:[TransferStatus,2];8:[TOrder,1];9:[ActID,1]</workflowCard>
    <taskCard>1:[TOrder,2];2:[CorpCode,%v_Corp_code%]</taskCard>
  </workflow>
</operateAction>
```

changeSignSQL

修改状态 SQL，不返回结果集

```
<changeSignSQL>
update dps_zy_client set flag = '5' where corp_code = '%v_Corp_code%' and client_code =
'%v_Client_code%'
</changeSignSQL>
```

affectSQL

相关影响操作 SQL，不返回结果集

```
<affectSQL>update zm_gl_caseinfo
set Case_worth='%v_Case_worth%'
where Corp_code='%v_Corp_code%' and Case_id='%v_Case_id%'
</affectSQL>
```

logSQL

事务记录 SQL，不返回结果集(保留)

refAction

其它相关的操作.用于调用事务服务（保留）

workflow

工作流调用
一个 commander 只能对应一个 workflow。一个 workflow 只能对应一个 workflowCard，但是可以对应 N 个 taskCard。

属性表

属性名	必要性	说明	默认值
workflowCard	必需	数据流内容对象，传递给工作流引擎以驱动整个工作流程，完成指定的操作。	无

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

taskCard	可选	任务卡。分发任务时使用，用来指定分发的部门、人员	false
----------	----	--------------------------	-------

workflowCard

数据格式: 标准格式

支持的参数名:

参数名	类型	实际意义	必需	说明
ProjectID	varchar(10)	工程标志	是	
CorpCode	varchar(10)	单位代码	是	
DocID	char(20)	公文编号	是	
StaffID	varchar(20)	职员编号	否	
DocType	varchar(60)	公文类型	否	
TOrder	int	工作序号	否	当前工作序号
DocTitle	varchar(100)	标题	否	
GroupID	int	组 ID 号	否	
Controller	varchar(20)	管理员	否	
TransferStatus	int	调用标志	是	只能取数字 0,1,2,3,4。 0：新建工作流 1：更改公文状态（公文状态为新邮件时改为收件箱状态，状态为新闻知公文状态时则改为阅知箱状态） 2：用户操作（提交、回退、删除、终止） 3：获取权限 4：更改管理员
ActID	int	动作编号	是	参见表 wfs_ActionDefine TransferStatus 为 2 时必需有 ActID。
DepartID	varchar(20)	部门代码	否	
DocLevel	int	公文级别	否	
UrgencyLevel	int	紧急级别	否	

公文状态：0 新邮件；1 收件箱；2 在办箱；3 催办箱；4 发件箱；5 垃圾箱；6 伴随删除；10 新闻知公文；11 阅知箱；20 已退回；21 伴随退回；100 完成；200 终止；201 伴随终止。

taskCard

数据格式: 标准格式

支持的参数名:

参数名	类型	实际意义	必需	说明
AddStaff	varchar(10)	增加职员	是	职员代码
TOrder	varchar(10)	工作序号	是	下一个要分发的工作序号
DepartID	char(20)	部门代码	否	
ActionControl	varchar(20)	职员编号	否	

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

signExpression

状态表达式 select flag as v_NoteFlag from ... where ...
返回结果集

```
<signExpression>select '10' as v_NoteFlag</signExpression>
```

succeedMessage

执行成功提示，如：新增成功！

```
<succeedMessage>申请成功！</succeedMessage>
```

lostMessage

执行失败提示，如：新增失败！

```
<lostMessage>申请失败！</lostMessage>
```

popParameter

抛出参数定义，向创建者发射变量的改变,在被调用的应用中定义

格式：1：[output, %v_valueClassName%]

示例说明：有一个弹出窗口用于查看物品信息，同时把所选的物品传出，用 radio 类型网格显示，当前行的改变触发一个按钮，配置如下：

当前行改变：

```
<relations_Linkage>  
  <initiative>grid_Radio</initiative>  
  <commanderName>comm_radio</commanderName>  
</relations_Linkage>
```

按钮的定义：

```
<commander>  
  <name>comm_radio</name>  
  <enabled>true</enabled>  
  <popParameter>1:[output,%v_goodsCode%]</popParameter>  
</commander>
```

talkAction

对话操作定义

子元素：

子元素	类型	必要	多个	说明	默认值
objectName	String	是	否	textMatrix.name 或 oneLineEdit.name	无
actionName	Enumerate	是	是	对话操作类型： actionName Enumerate	无

objectName

textMatrix.name 或 oneLineEdit.name

actionName

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

值	描述
AddRow	编辑 textMatrix 时使用，在末尾增加一行
InsertRow	编辑 textMatrix 时使用，当前行中插入一行
DeleteRow	编辑 textMatrix 时使用，删除当前行
Clear	编辑 textMatrix 和 oneLineEdit 时使用，清空
DeleteData	编辑 textMatrix 时使用
Refresh	编辑 textMatrix 时使用，刷新
PageHome	编辑 textMatrix 时使用，首页
PageUp	编辑 textMatrix 时使用，上一页
PageDown	编辑 textMatrix 时使用，下一页
PageEnd	编辑 textMatrix 时使用，最后一页
PageTo	编辑 textMatrix 时使用，到第 n 页
Update	编辑 oneLineEdit 时使用，使主键不可用
NewRow	编辑 oneLineEdit 时使用，新增记录
Quit	退出（只限于 application）关闭工作区页面
Exit	退出（只限于 application）关闭弹出页面
SelectAll	Gird 中 check 操作
SelectClear	Gird 中 check 操作
SelectBack	Gird 中 check 操作

noteOperatedState

状态位定义

子元素：

子元素	类型	必要	多个	说明	默认值
signValueName	String	是	否	一个 valueClass 实例的名称，此实例标识了凭证的状态位	无
sign	Sign	是	是	凭证状态描述定义	\

```
<noteOperatedState>
  <signValueName>v_NoteFlag</signValueName>
  <sign>
    <value>10</value>
    <objectName>client_code</objectName>
    <enableKey>>false</enableKey>
    .....
  </sign>
  .....
</noteOperatedState>
```

signValuName

一个 valueClass 实例的名称，此实例标识了凭证的状态位

```
<signValueName>v_NoteFlag </signValueName>
```

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

sign

凭证状态描述定义

子元素：

子元素	类型	必要	多个	说明	默认值
value	String	是	否	状态值	无
objectName	String	是	是	网页对象名	无
enableKey	Boolean	是	是	对象是否可用	无

value

状态值
default

objectName

网页对象名

enableKey

true 或者 false.对象是否可用。

keyDefine

子元素：

子元素	类型	必要	多个	说明	默认值
tabKey	Long	否	否	键盘的 ASCII 码，切换焦点用，类似于 tab 按钮，但它只作用于 operationEdit.jsp 要与 nextField 结合使用	无
hotKey	HotKey	否	是	热键操作	\

```

<keyDefine>
  <hotKey>
    <key>32</key>
    <fieldName>Address</fieldName>
    <commanderName>b_queryaddress</commanderName>
  </hotKey>
</keyDefine>

```

tabKey

键盘的 ASCII 码，切换焦点用，类似于 tab 按钮，但它只作用于 operationEdit.jsp 要与 nextField 结合使用

hotkey

热键操作

子元素：

元素名	类型	必要	多个	说明	默认值
key	Long	是	否	键盘的 ASCII 码	无

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

fieldName	String	否	否	如果与焦点有关，那么指定的焦点位置	无
commanderName	String	是	否	点击的按钮的名称	无

key

键盘的 ASCII 码

fieldName

如果与焦点有关，那么指定的焦点位置

commanderName

点击的按钮的名称

5. 配置文件中的枚举类型说明

fieldType Enumerate 类型：

值	描述
long	整型
double	浮点型
boolean	布尔型
String	字符串型
datetime	日期型

editType Enumerate 类型：

值	描述
Text	同 html 定义
Password	同 html 定义
Select	同 html 定义
Textarea	同 html 定义
Checkbox	同 html 定义
Radio	同 html 定义
Hidden	同 html 定义
Date	日期型，在网页中使用 Text

gridStyle Enumerate 类型：

值	描述
none	普通的网格
radio	可以选择单行的网格
check	可选择多行的网格

linkageType Enumerate 类型：

值	描述
Query	查询，主要用于查询结果集，并填充 field 的值
Restrict	限制一个下拉框的填充数据，与其对应的 functionString 是一个同 Field.fillListSQL 相同的查询语句（结果集结构相同）
Caculate	计算，主要用于相关 field 或 valueClass 的计算关联
StringPlus	字符串相加，主要用于相关 field 或 valueClass 的字符串相加，形如：

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>

	Client code := %DocID%
FormatControl	格式化字符串处理，他与 GetIndexString，LoopGetIndexString，GetDetailString，GetPartString 配合使用

注意：在 Query 类型中，结果集的命名要与 field 的 name 保持一致，如不同，可定义别名（在 select 中）

condition Enumerate 类型:

值	描述
Not Null	不能为空
Unique	数据库中唯一
Not Exist	存在检查（数据库中必须存在）
Logic Check	逻辑检查

saveSQL Enumerate 类型:

值	描述
UpdateSQL	调用 object.updateSQL
InsertSQL	调用 object.insertSQL
DeleteSQL	调用 object.deleteSQL

actionName Enumerate 类型:

值	描述
AddRow	编辑 textMatrix 时使用，在末尾增加一行
InsertRow	编辑 textMatrix 时使用，当前行中插入一行
DeleteRow	编辑 textMatrix 时使用，删除当前行
Clear	编辑 textMatrix 和 oneLineEdit 时使用，清空
DeleteData	编辑 textMatrix 时使用
Refresh	编辑 textMatrix 时使用，刷新
PageHome	编辑 textMatrix 时使用，首页
PageUp	编辑 textMatrix 时使用，上一页
PageDown	编辑 textMatrix 时使用，下一页
PageEnd	编辑 textMatrix 时使用，最后一页
PageTo	编辑 textMatrix 时使用，到第 n 页
Update	编辑 oneLineEdit 时使用，使主键不可用
NewRow	编辑 oneLineEdit 时使用，新增记录
Quit	退出（只限于 application）关闭工作区页面
Exit	退出（只限于 application）关闭弹出页面
SelectAll	对 check 的 grid 操作
SelectClear	对 check 的 grid 操作
SelectBack	对 check 的 grid 操作

windowType Enumerate 类型：

值	描述
None	普通类型，需要读取 top, left, height, width
FillScreen	全屏幕窗口，不需定义 top, left, height, width

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	Edi tDate: <6/21/2003>

Center	居中窗口，需要读取 height, width
noPOP	直接的本页提交，不是弹出口

6. 附录

6.1 如何使用自定义标签 UserLabel?

- 1) 自定义标签 UserLabel 在元素 Field 中定义，有 Field 元素的通用属性。**不同的是**其 editType 为 UserLabel,并且整个 UserLabel 标签(<UserLabel name = "pic1">)将会被常量或者变量值(valueClass 等)代替。而普通的 Field 只有 value 值会被代替。

- 2) 示例：

- i. HTML 中的语句：<UserLabel name = "pic1"> (**大小写不敏感！**)

- ii. XML 中的定义：<field> (**大小写敏感！**)

```

<name>pic1</name>
<fieldType>String</fieldType>
<editType>UserLabel</editType>
<initValue>
  <![CDATA[
    <img src = "pic1">
  ]]
</initValue>
或者(<format>select pic from zm_zd_user where ClientID =
'%ClientID%'</format>)
</field>

```

6.2 对页面上显示特定图片的使用说明

譬如要显示某个客户的照片在 HTML 页面上，以用于查看或者打印。那么步骤如下：

- 1) 扫描某个客户的照片后将照片按一定的规则命名。
- 2) 将扫描后的照片保存在指定目录的文件夹中。
- 3) 对于 WebSphere 服务器,必须保存于路径 WebSphere\installedApps\hostname\MisExpress_war.ear\MisExpress.war\ 目录下。譬如 WebSphere\installedApps\hostname\MisExpress_war.ear\MisExpress.war\ClientImages\目录。
- 4) 对于 Tomcat 服务器，必须保存于路径 tomcat\WebApps\MisExpress\目录下，譬如 tomcat\WebApps\MisExpress\ClientImages\。如果 Tomcat 中无 tomcat\WebApps\MisExpress\是因为平台程序未被设置成解压运行方式。可将 tomcat\conf\Server.xml 中的 unpackWARs="false" 改为 unpackWARs="true"

警告：重新部署平台文件(MisExpress.war)时，注意要将照片文件备份，否则重新部署后，其照片会被自动删除。部署后应将备份的照片文件恢复到先前的目录中。

推荐的做法是先将照片扫描到备份文件夹中。然后将备份文件夹中的文件 copy 到指定目录，而不是将照片扫描后直接放入到指定目录的文件夹中而不进行存档！

- 5) 在数据库中对图片的路径进行描述，并与其它数据建立联系。譬如如果是客户的照片，那么客户信息表中必须应该有一个字段，如 pic 等。用以描述**显示图片的 HTML 代码**。如果将某个 ID 为 0001 的客户的照片保存于路径：WebSphere\installedApps\hostname\MisExpress_war.ear\MisExpress.war\ClientImages\0001.jpg。那么 pic 的值应该为 。**注意：**必须有"../"并且包括尖括号。
- 6) 在 HTML 文件中需要显示图片的写如下代码：<UserLabel name = "pic1">。必须有

<MisExpress 业务解释平台>	Version: <1.0>
配置文件编写说明文档	EditDate: <6/21/2003>


name,其 name 与 XML 中 Field 的 name 一致。必须有单词 UserLabel。<UserLabel name = "pic1"> 将被替换成标准的 HTML 代码, 譬如 ``, 此 HTML 源代码可存在数据库中。

在 XML 文件中的定义如下:

```
<field>
  <name>pic1</name>
  <fieldType>String</fieldType>
  <editType>UserLabel</editType>
  <format>select pic from zm_zd_user where ClientID =
'%ClientID%'</format>
</field>
```

7) 结束! 当页面被打开时, HTML 页面上应该会显示此图片。

6.3 如何增加 URL 连接以弹出详细信息页面

 应用范围: 譬如页面上显示某个供应商, 单击这个供应商会弹出一个页面显示这个供应商的详细信息。

说明: 弹出页面的过程中需要传递主键, 目前只针对传递一个主键做过测试, 测试通过。

示例:

譬如需要在**网格**中对以下标签增加 URL 连接:

```
<td width="18%" align="center" height="10" bgcolor="#E4F3EC"><userlabel
name="g_SupplyMember"></td>。
```

g_SupplyMember 指供应商名称。点击供应商名称弹出一个页面显示显示此供应商的详细信息

步骤:

1) 在需要增加 URL 连接的字符的后面加上标签, 针对当前示例加入的标签位置如下:

```
<td width="18%" align="center" height="10" bgcolor="#E4F3EC"><userlabel
name="g_SupplyMember"></a></td>。
```

2) 在需要增加 URL 连接的字符的前面加上字符串:`<a target="_blank" href="../centercontrol?messageString=1:[appName,FacInfo];2:[projectId,yyProject];3:[className,DataPond];4:[MemberID,<userlabel name="g_MemberID">]">` 这个字符串中包含 userlabel, 以存放变量的值, 譬如主键或者平台要求的变量的值 (projectId 等, 此处为了方便已将 projectId 固定)。当前示例中只传递一个主键 **g_MemberID**。如果有多个主键, 可写多个 userlabel 进行值传递(多个传递未测试!)。

3) 在父页面的 XML 文件中定义一个 Field: **g_MemberID**。这个值一般来源于数据库。

4) 定义一个弹出页面 (是一个新的应用, 只需要将此应用定义在 fyl_PageInfo 表中, 可不在 fyl_SubjectDefine 中描述), 以接收传递的参数: **g_MemberID**。在当前示例中, g_MemberID 的值由变量(ValueClass)**MemberID** 接收。所以定义一个名称为 MemberID 的 ValueClass。

5) 将定义的 ValueClass: MemberID 作为查询条件, 查询出当前供应商的详细资料。

6) OK!

6.4