广西院合同管理合同实例设计

V1.0

重庆百木科技有限公司

2017.02.04 By mengzq

文档控制

文档更新记录

| 日期 | 更新人 | 版本 | 备注 |
| --- | --- | --- | --- |
| ??? |  |  |  |
| 2017-2-13 | mengzq |  | 修改合同属性的Range值，添加Range的国际化设置；添加内容合同管理需求-添加合同信息字段 |
| 2017-2-14 | mengzq |  | 更新一些截图 |
| 2017-2-17 | mengzq |  | 增加Policy的state的国际化 |
| 2017-7-76 | Albert |  | 根据访谈后的需求更新本文档，调整字段及命名 |

文档审核记录

| 日期 | 审核人 | 职务 | 备注 |
| --- | --- | --- | --- |
|  |  |  |  |
| 2017/7/6 | Albert | 项目经理 | 审核更新本文档、调整部分排版 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

广西院合同管理需求内容：

合同管理流程：

* 主要监控合同流转过程状态改变
* 状态变化后消息通知相关干系人
* 统计合同所属项目的信息，尤其关注合同经营产值的变化情况

A.系统中增加合同信息字段：

系统中已经用route配置好流程，在Enovia的route流程中，无法通过OOTB配置增加定义字段，需要修改后台的数据模型，以实现特定属性字段的增加。

说明：本实例所有schema\组件\JSP\JPO等.都以Bmbim\_contract 为前缀，便于统一查询。

Schema

Agreement

Bmbim\_contract\_items

Bmbim\_remain\_amount

Bmbim\_contract

Bmbim\_contract

Bmbim\_contract\_num

Bmbim\_contract\_name number

Bmbim\_client\_name

Bmbim\_contract\_attr

Bmbim\_process\_comment

Bmbim\_prepayment

Bmbim\_contract\_type

Bmbim\_total\_amount

Bmbim\_contract\_payment

Bmbim\_contract\_enddate

Bmbim\_our\_sign\_date

Bmbim\_client\_sign\_date

Terminate

Transfer

Dismiss

Modify

Working

Schema说明

**Type**

Bmbim\_contract 合同

**Attribute(空格用下划线代替)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bmbim\_contract\_num | 合同号 | 字符串 | string |  |
| Bmbim\_contract\_name | 合同名称 | 字符串 | string |  |
| Bmbim\_contract\_clientname | 业主名称 | 字符串 | string |  |
| Bmbim\_contract\_type | 合同分类 | 字符串 | string | 规划、测量、港口、航道和枢纽 |
| Bmbim\_contract\_processfee | 合同进度款 | 数值 | real | 提醒功能 |
| Bmbim\_contract\_clientsign | 业主签定时间 | 时间 | date |  |
| Bmbim\_contract\_signdate | 我院签定时间 | 时间 | date |  |
| Bmbim\_contract\_deliverylist | 交付清单及时间及金额 | 清单 | String |  |
| Bmbim\_contract\_deliverydate | 交送日期 | 时间 | date |  |
| Bmbim\_contract\_received | 收款金额 | 数值 | real |  |
| Bmbim\_contract\_tracerecord | 收款记录 | 字符串 | string |  |
| Bmbim\_contract\_totalamount | 合同总金额 | 数值 | real |  |
| Bmbim\_contract\_state | 合同状态 | 字符串 | string |  |

合同类型range:

规划Planing、测量Measuring港口Port, 航道SailChannel, 枢纽Hub

**Policy**

Bmbim\_contract(合同)  **Format[All]**

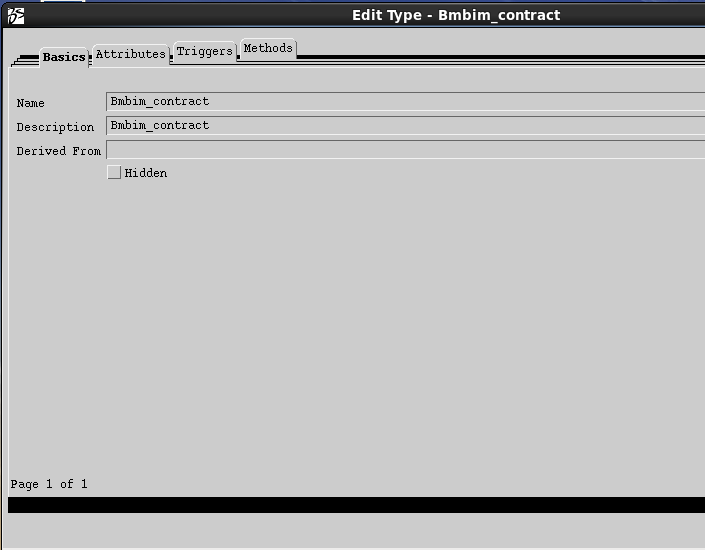
|  |  |
| --- | --- |
| 合同创建 | contract\_info\_create |
| 分院领导传阅 | contract\_branchleader\_review |
| 总院领导传阅 | contract\_generalleader\_review |
| 业主传阅 | contract\_client\_review |
| 签订 | contract\_official\_signed |
| 收款 | contract\_wait\_payment |
| 结束 | contract\_end |

# 一、建立AO

将上述合同管理的数据模型依次在Business中建立

注意：每次改动AO,必须在系统中重新加载高速缓存，并刷新页面，才能生效

1. 创建Type(Bmbim\_contract)

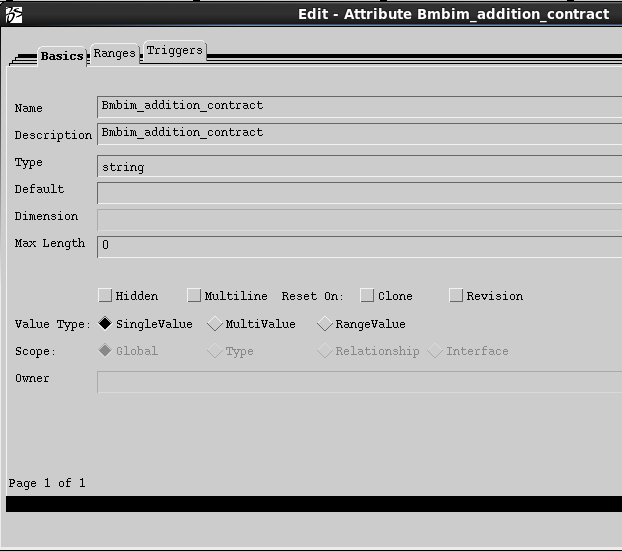


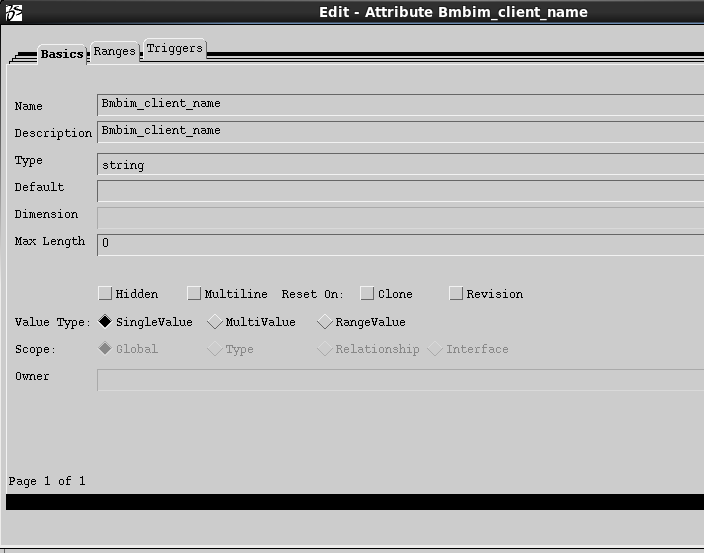
1. 创建Attribute(

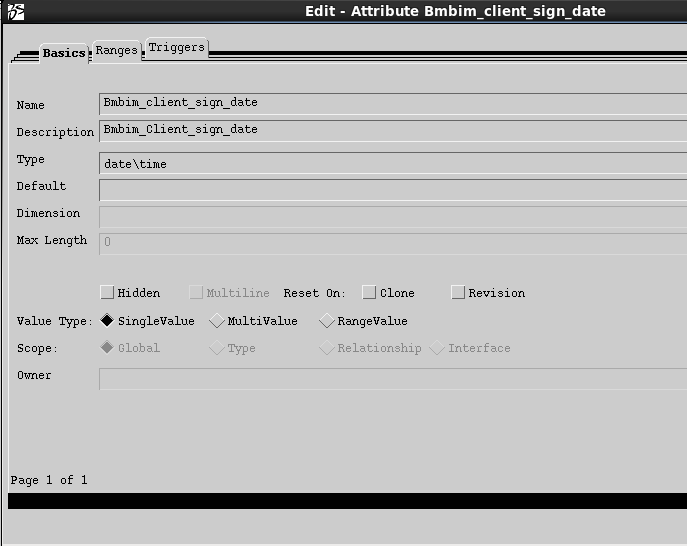
|  |
| --- |
| Bmbim\_contract\_num |
| Bmbim\_contract\_name |
| Bmbim\_client\_name |
| Bmbim\_contract\_attr |
| Bmbim\_client\_sign\_date |
| Bmbim\_our\_sign\_date |
| Bmbim\_contract\_enddate |
| Bmbim\_contract\_payment |
| Bmbim\_addition\_contract |
| Bmbim\_prepayment |
| Bmbim\_total\_amount |
| Bmbim\_remain\_amount |
| Bmbim\_process\_comment |
| Bmbim\_contract\_items |

)

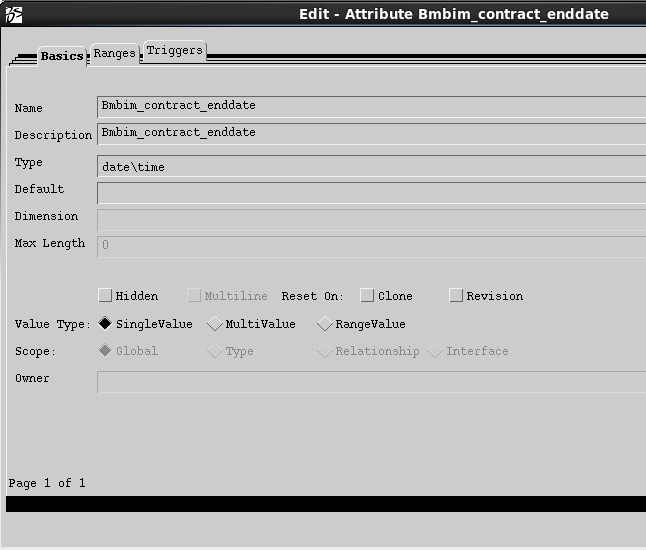
注意：必须在建立时选择Attribute的数据类型，如字符string、实数real .etc.，编辑时不能修改

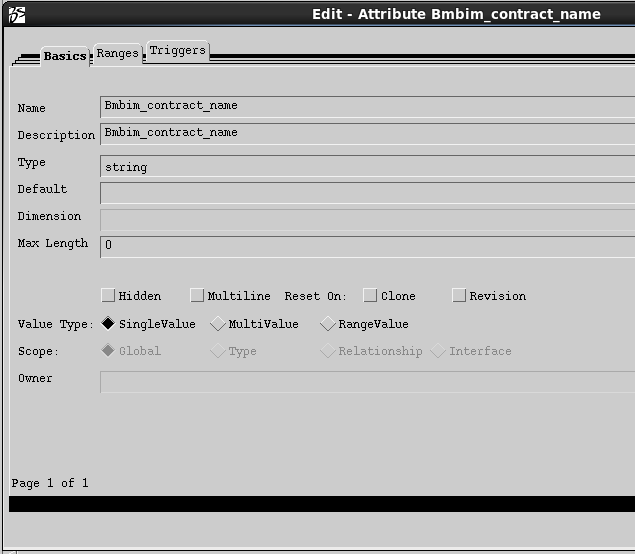


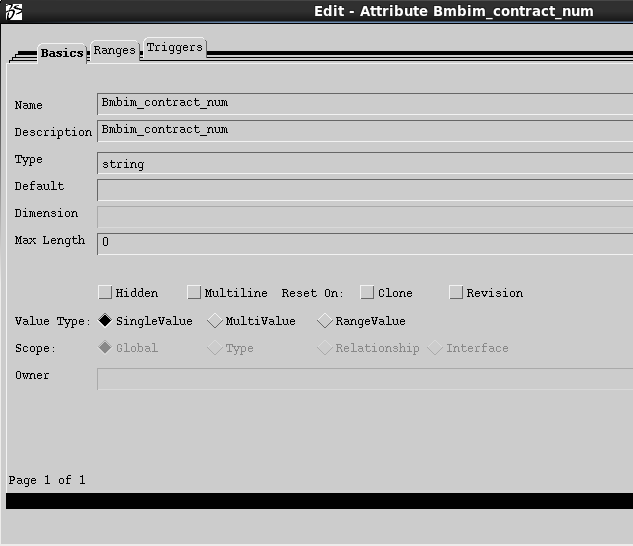


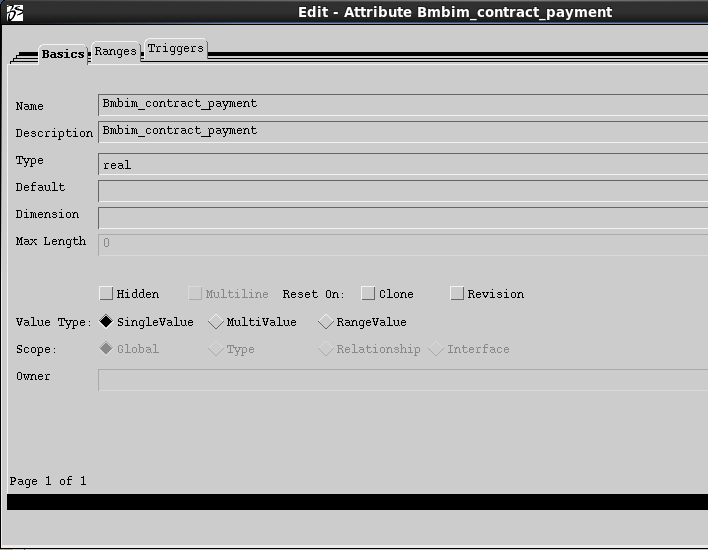




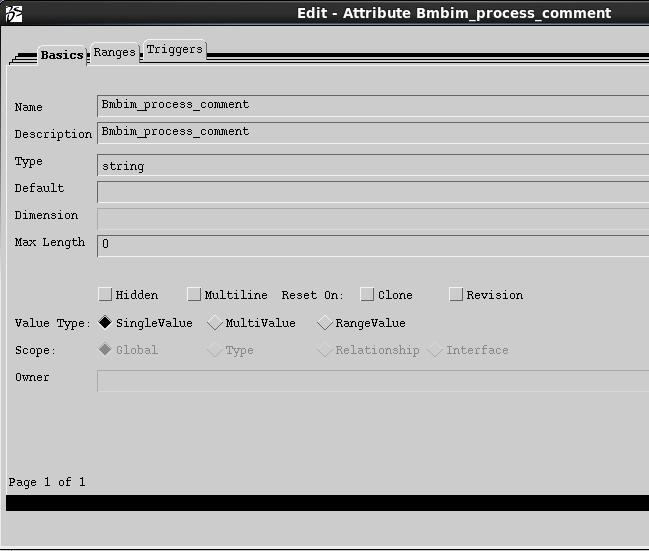


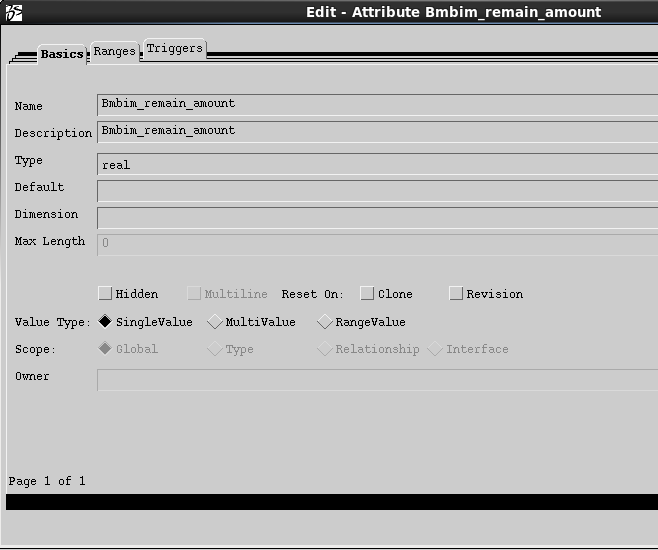


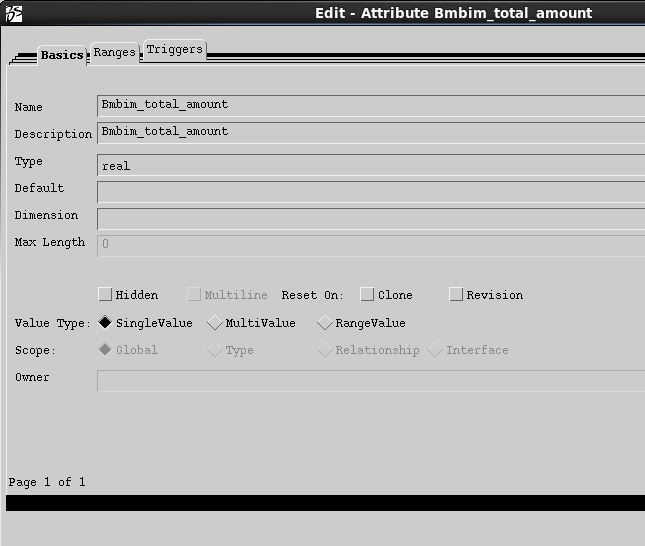


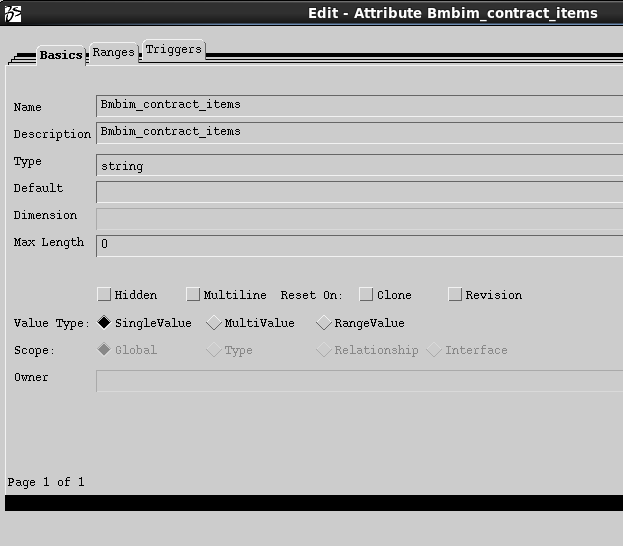


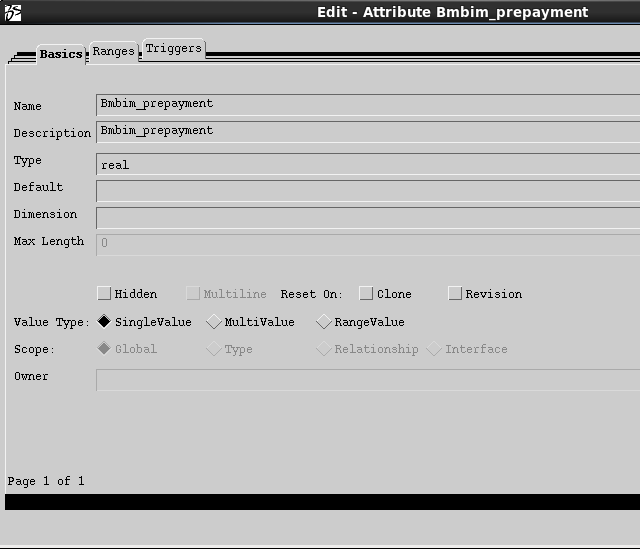




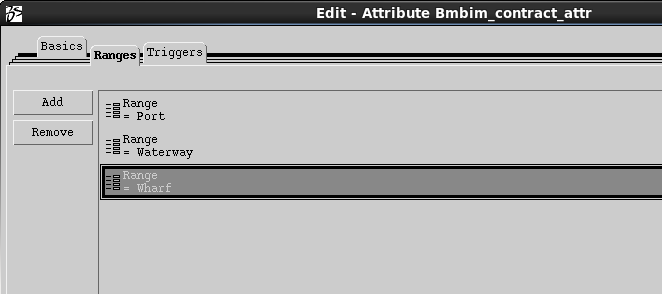




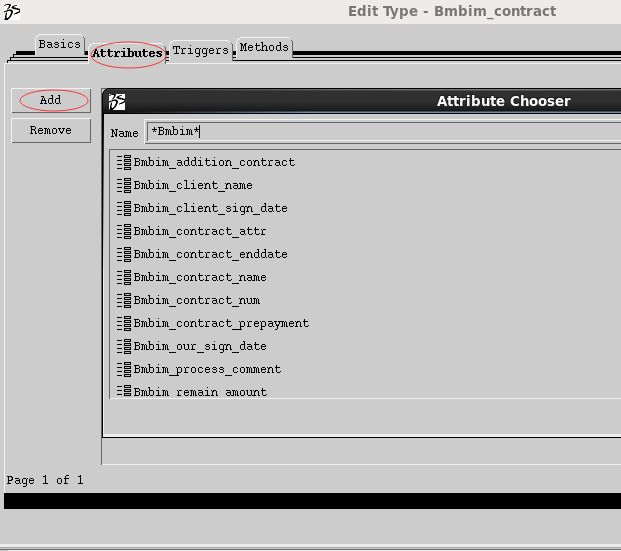




1. 配置Attribute(Bmbim\_contract\_attr)的range(Port,Waterway,Wharf)（Attribute有range时）



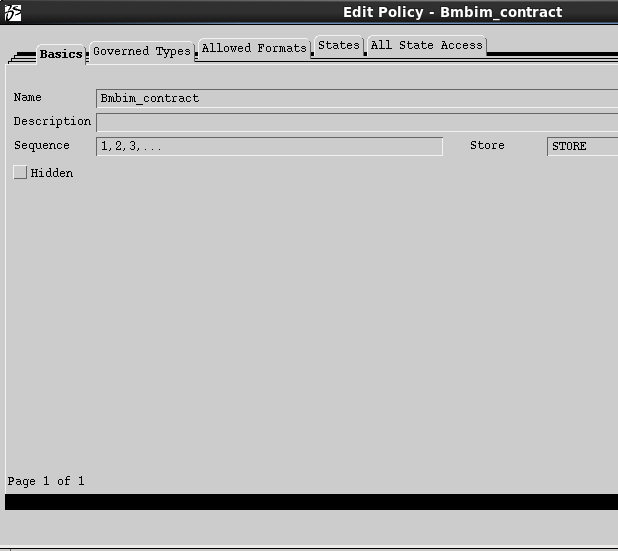
1. 将Attribute添加到Type(Bmbim\_contract)中





1. 建立关系
2. 将关系关联管理对象Type
3. 建立统治策略Policy(Bmbim\_contract)

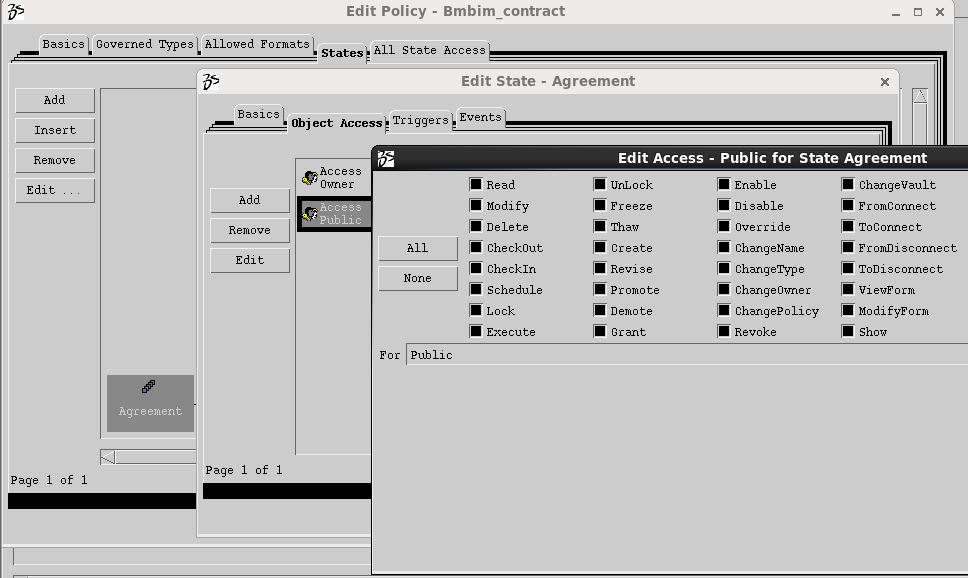
备注：要创建任意一个BO对象，则其AO中必须至少要拥有一个policy



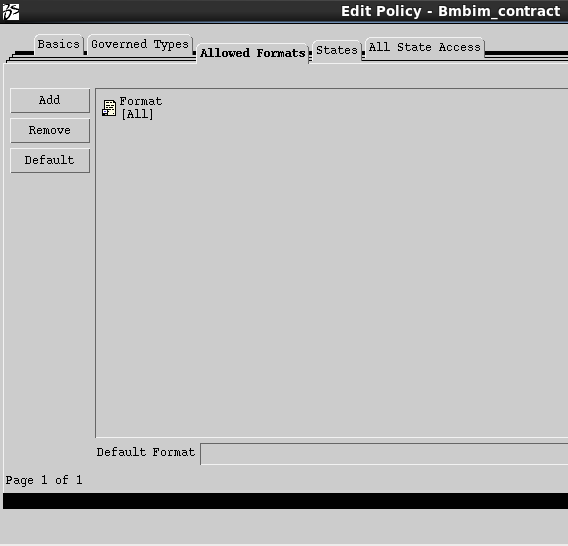
1. 配置policy(Bmbim\_contract)的生命周期状态state(

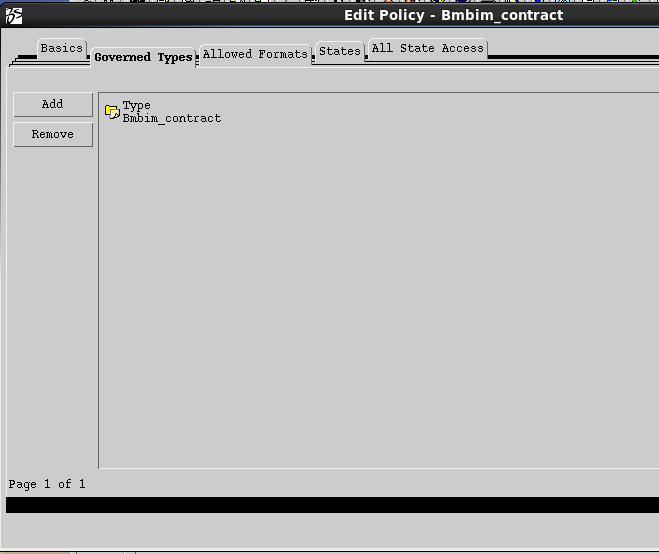
|  |
| --- |
| Agreement |
| Working |
| Modify |
| Dismiss |
| Transfer |
| Terminate |

)



备注：其中的”Access”是控制其实例BO在其状态的权限；一般策略都不需配置其自身上传文件格式，如有需求，则在”Allow Formats”中配置，如下图



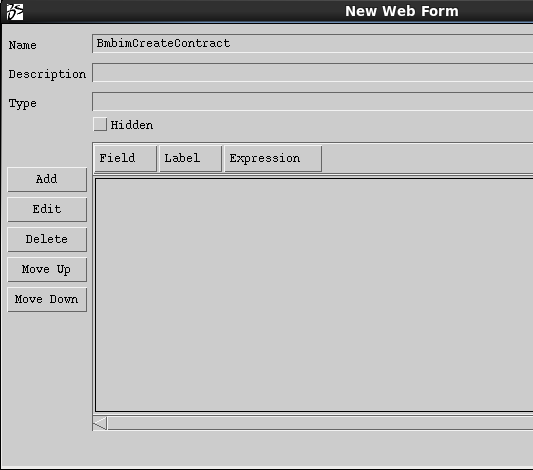


在系统中注册（注册参见文档下面内容 **五、注册**）上面创建的AO

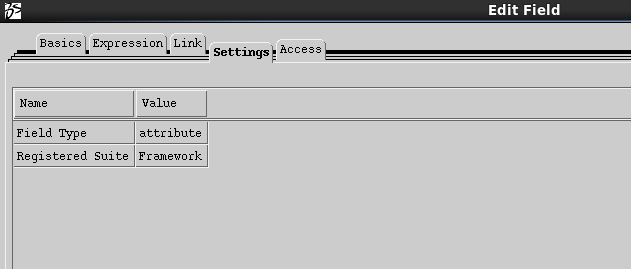
小结：建立AO的过程就是相当于在数据库中创建表、字段。

# 二、通过网页创建一个合同对象

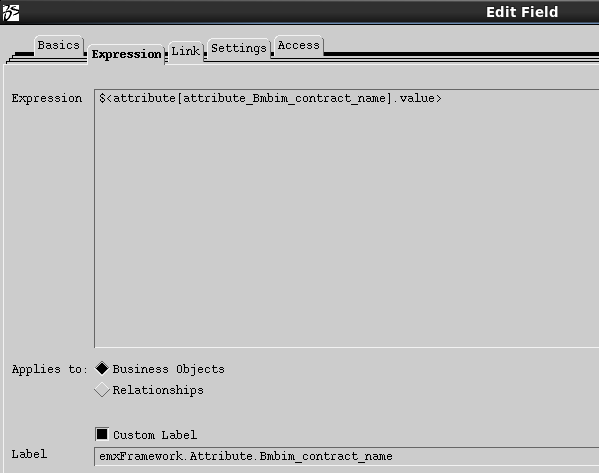
1.创建一个webform(BmbimCreateContract)



添加字段**合同名称**

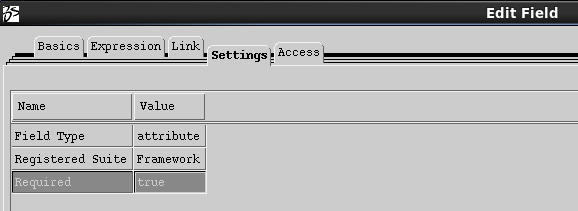


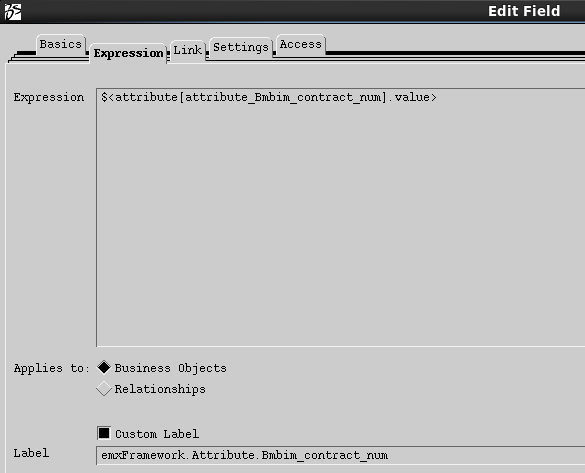
<Registered Suite,Framework>是国际化时添加的参数



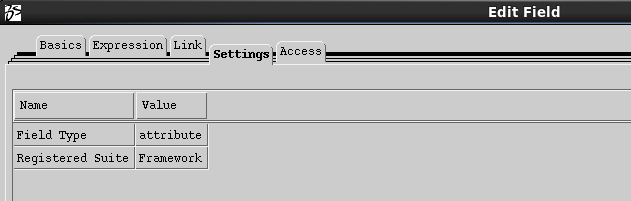
Label为字段的标记名，命名按国际化规则emxFramework.类型.注册名

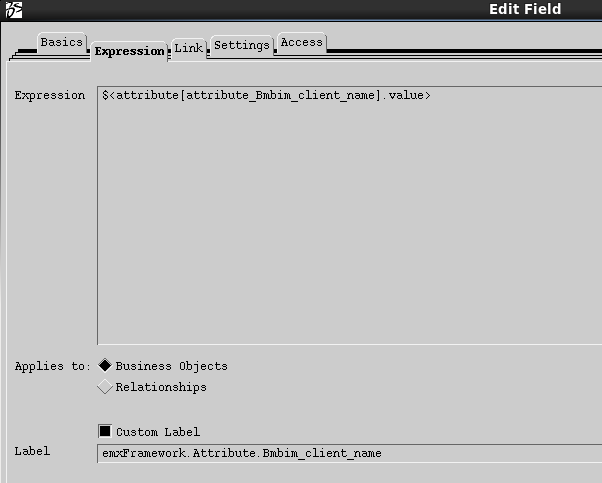
添加字段**合同号**





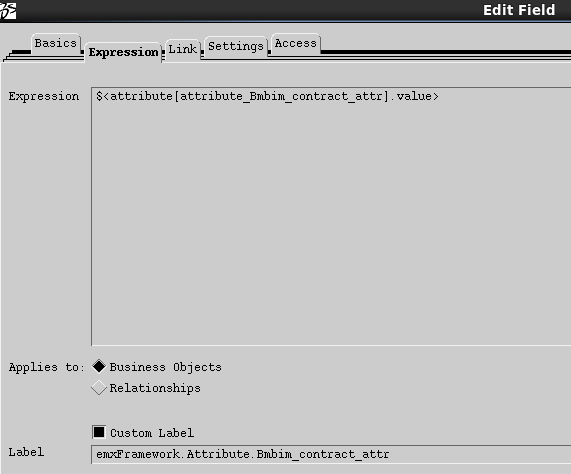
添加字段**业主名称**

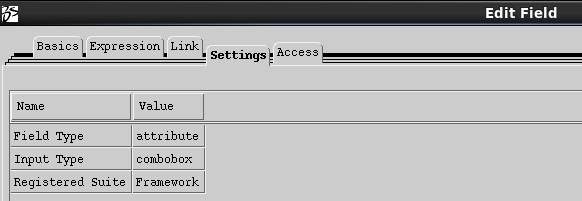




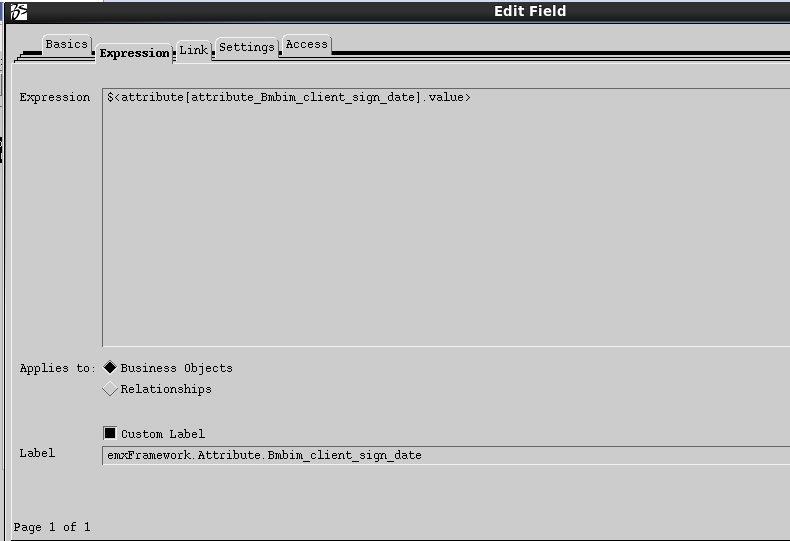
**附加：**

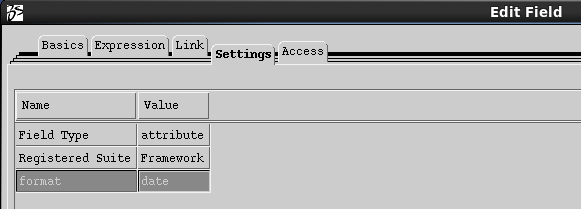
添加字段**合同属性**



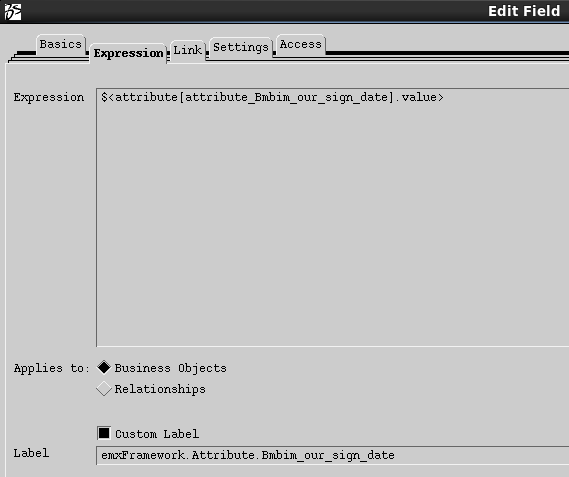


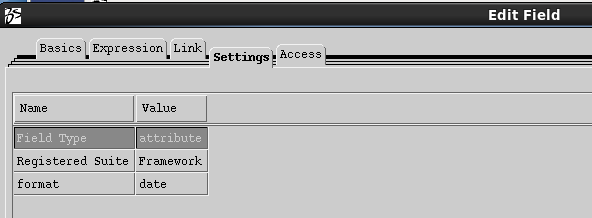
添加字段**业主签定时间**



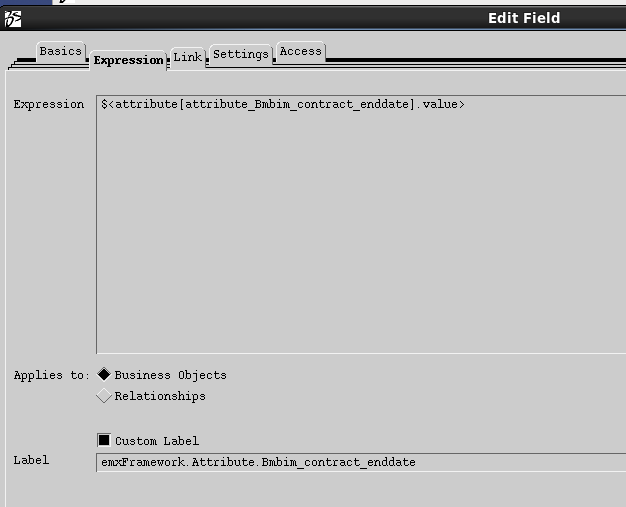


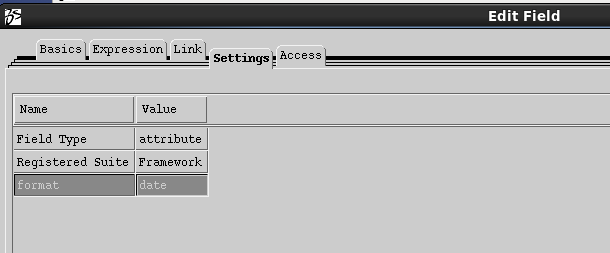
添加字段**我院签订时间**



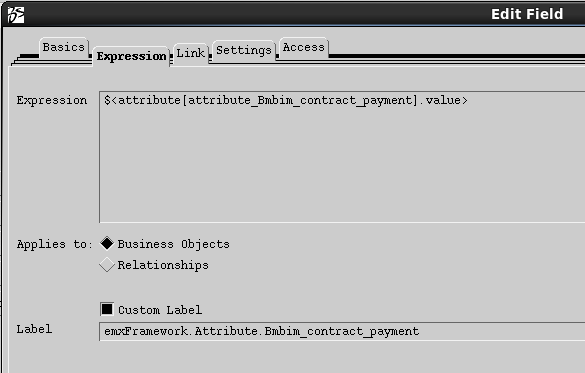


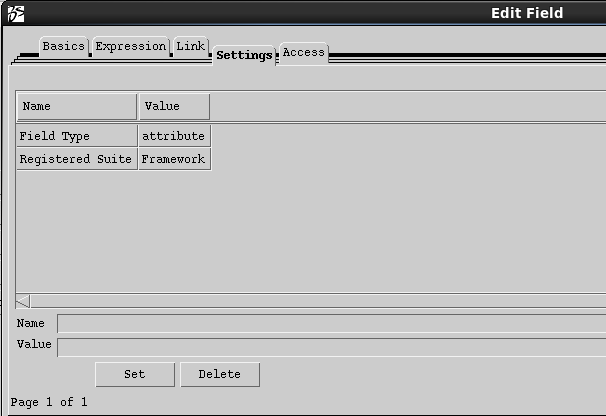
添加字段**文件交付清单及时间**



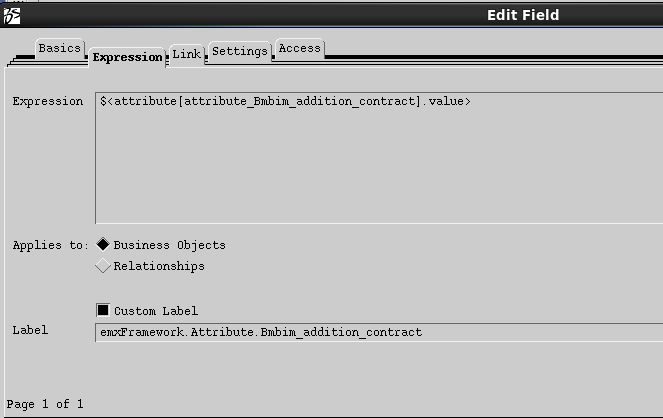


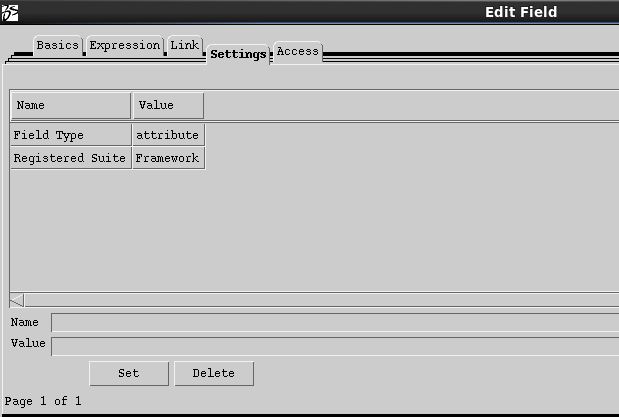
添加字段**合同回款**



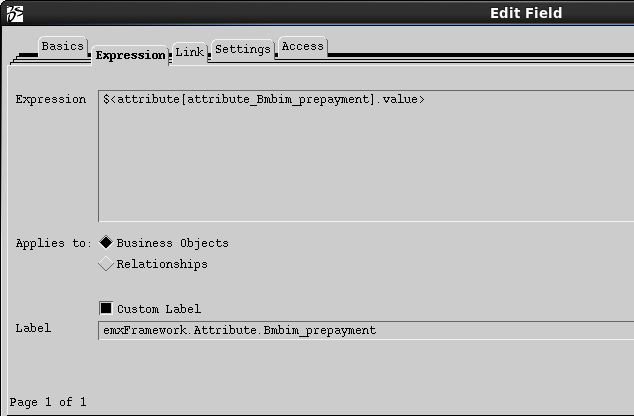


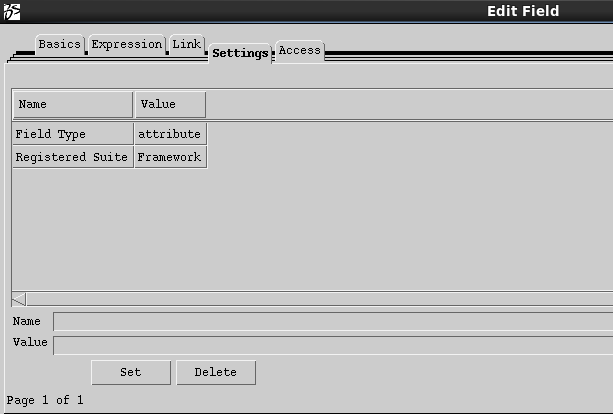
添加字段**补充合同**



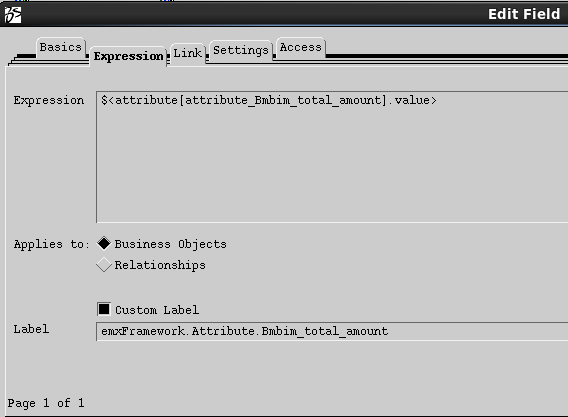


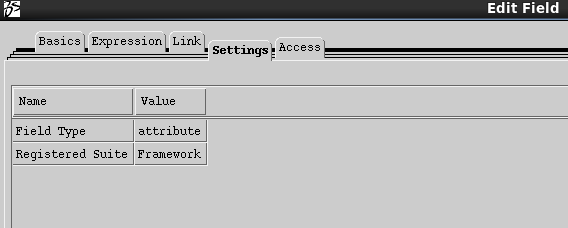
添加字段**收预付款金额**



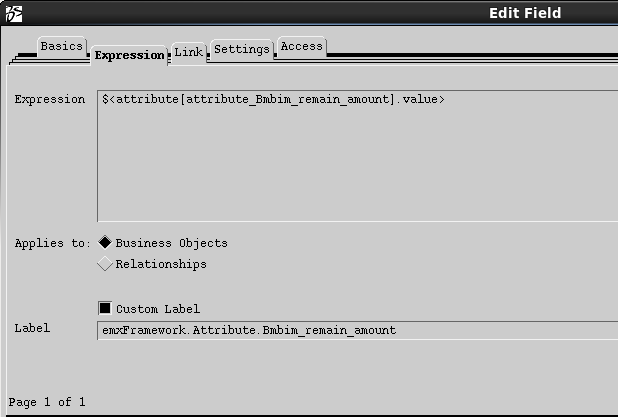


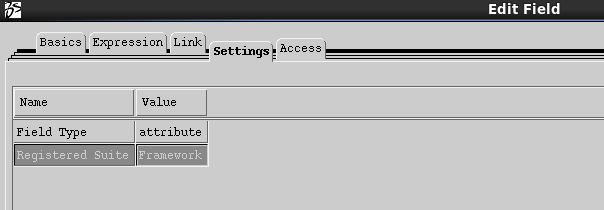
添加字段**合同总金额**



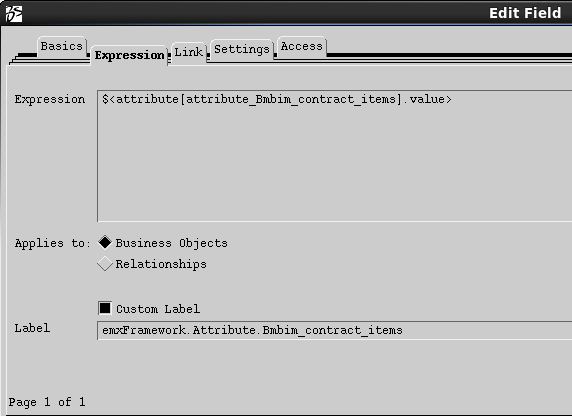


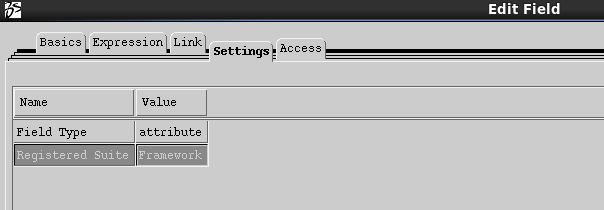
添加字段**剩余收款金额**



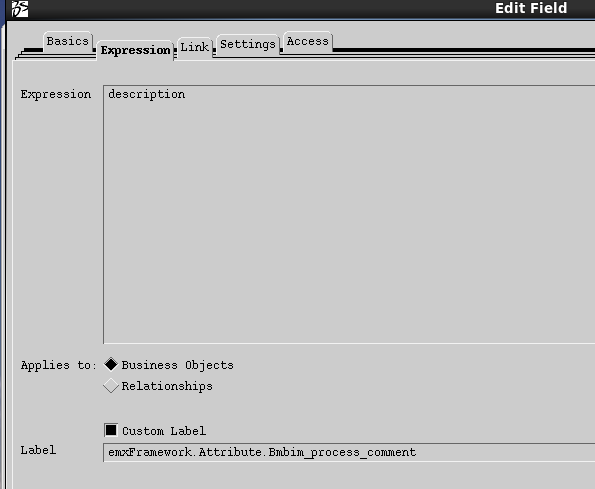


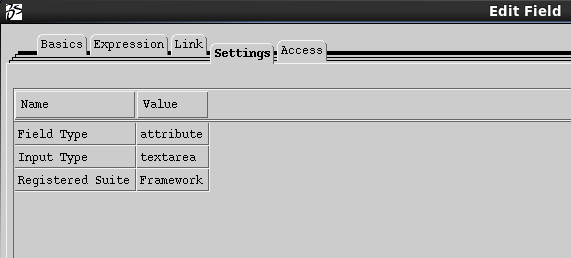
添加字段**合同清单**



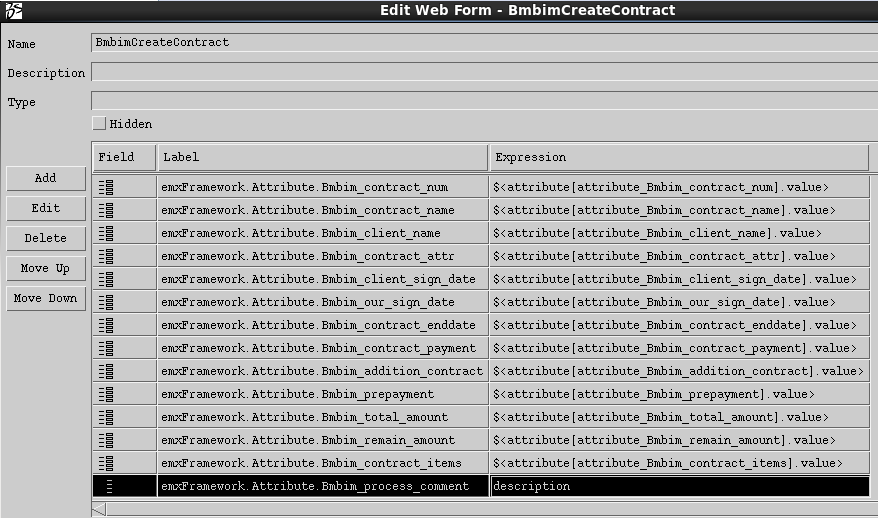


添加字段**合同流程过程中的批注**

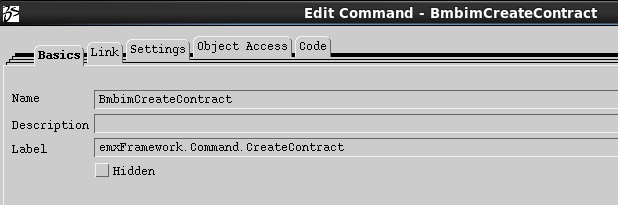




添加完字段后



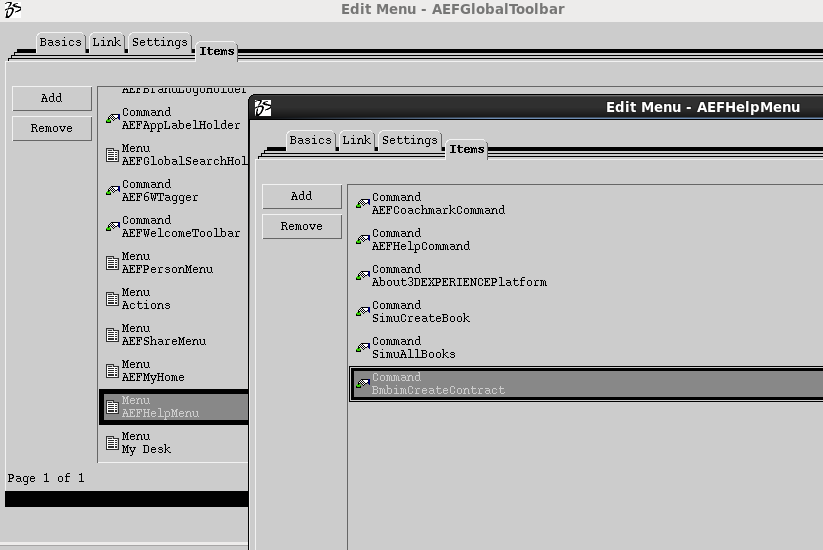
2创建一个command(BmbimCreateContract)





1. 在系统中关联此command(BmbimCreateContract)





1. 在系统中注册



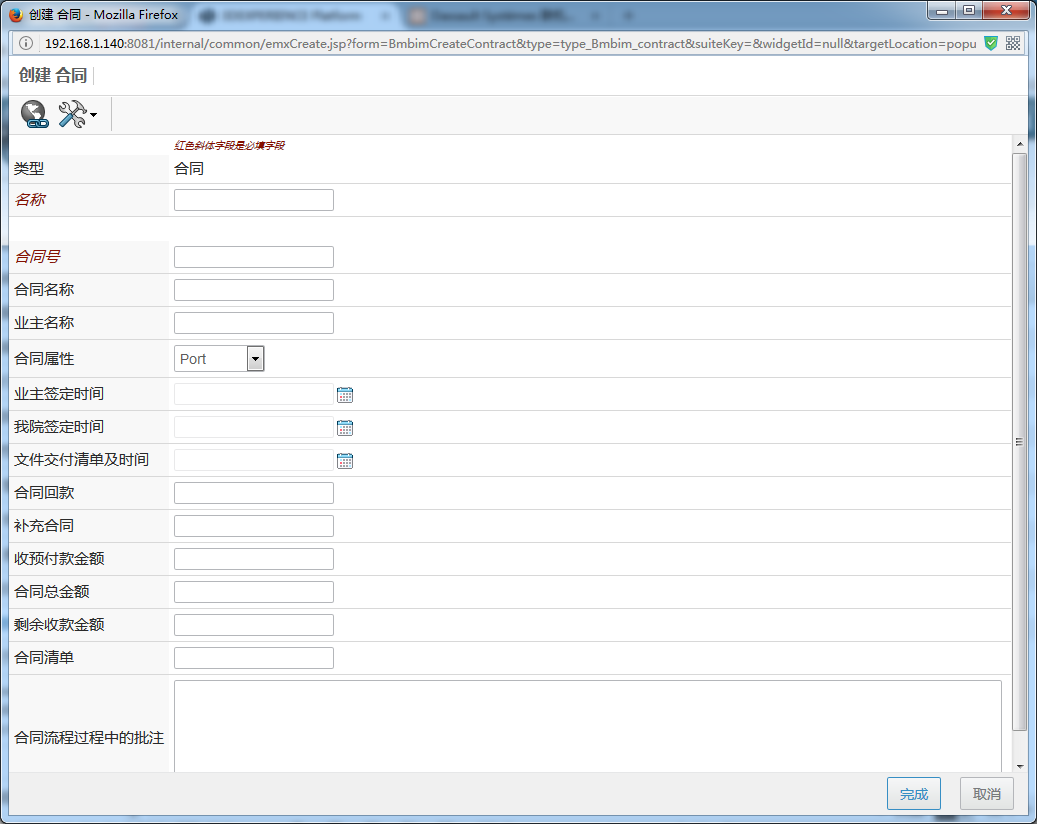
符号名称 type\_Bmbim\_contract

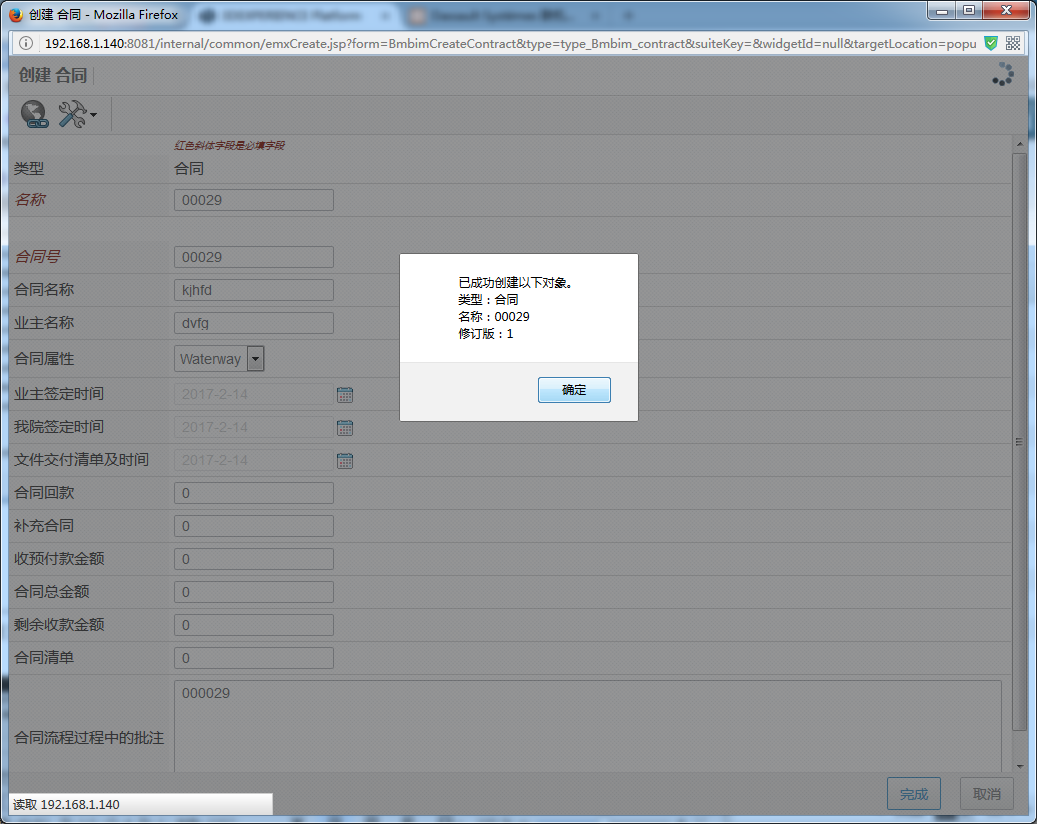
原始名称 Bmbim\_contract（原始名称的字符串若有空格用\_代替）

在系统中可以看到

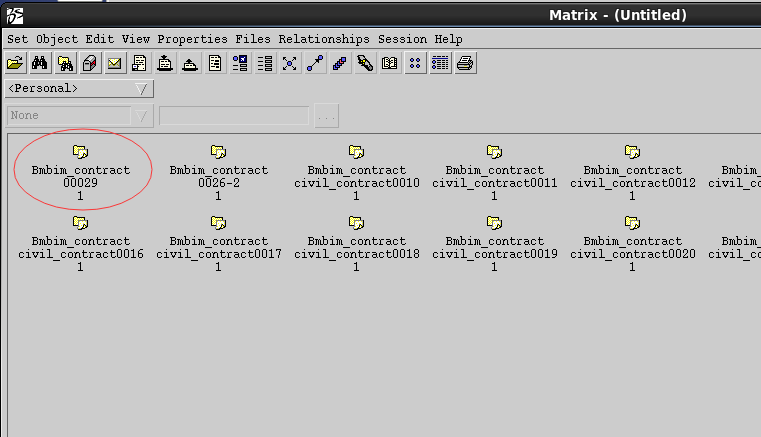


点击创建





在Maxtrix中查看

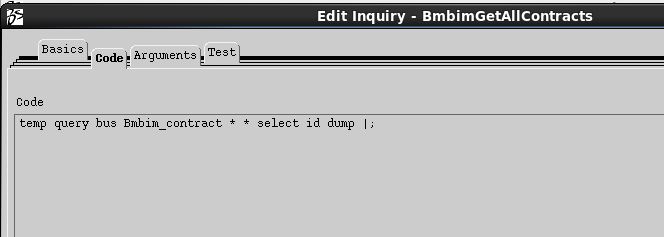


小结：UI组件webform的建立，webform,command,menu的基本参数的配置和宏的设置，实现能够通过网页创建合同对象。

# 三、查询并修改合同

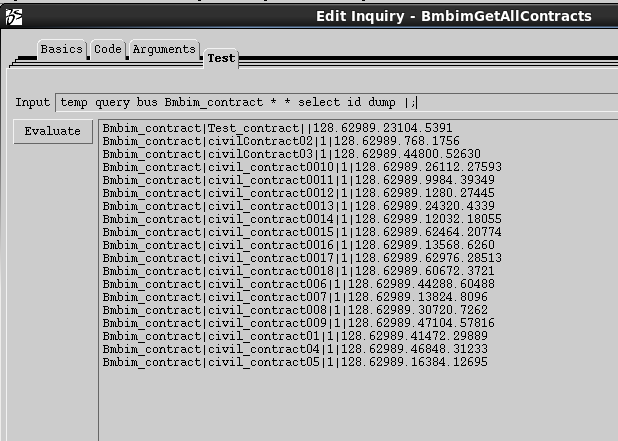
1.创建一个inquiry(BmbimGetAllContracts),查询合同信息



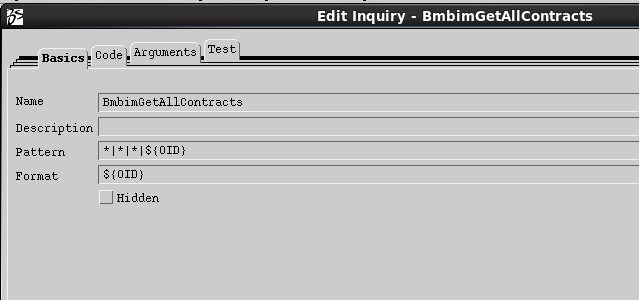


Code=temp query bus Bmbim\_contract \* \* select id dump |;

可以先在Test中进行测试



再设置输出格式



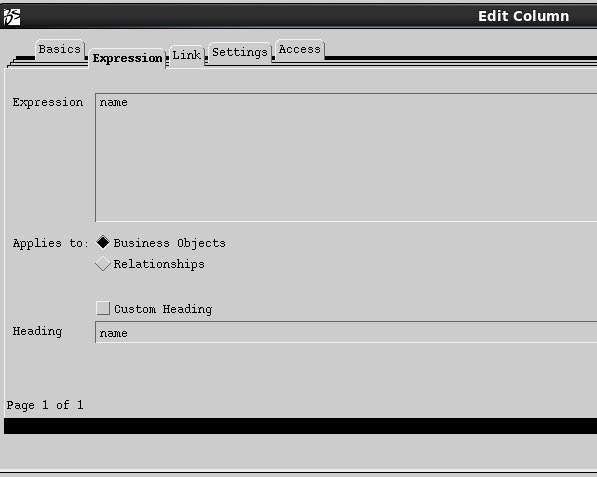
Pattern与Test测试出的数据匹配，每个BO都有两个标识符，1是tnr(type/name/revision),另一个是oid,相当于表的主键

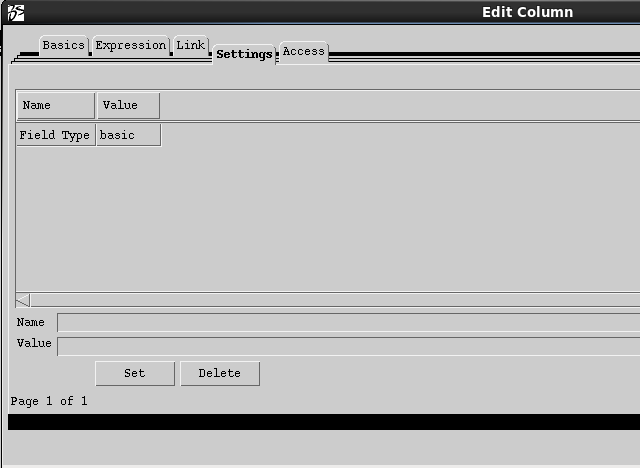
1. 创建一个table(BmbimAllContracts),展示合同信息



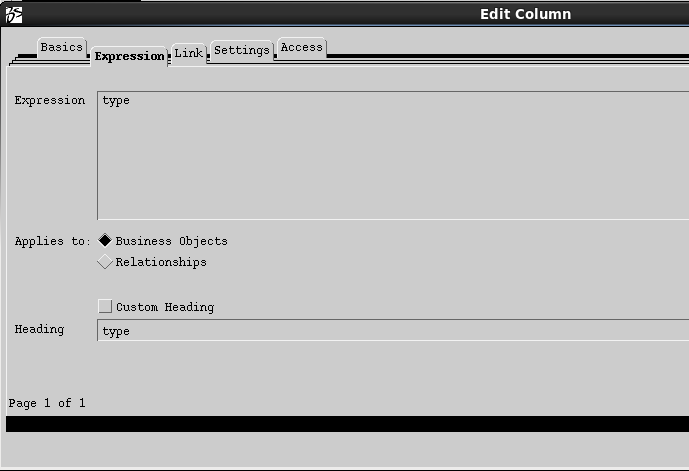
再增加一些列，展示合同的基本信息

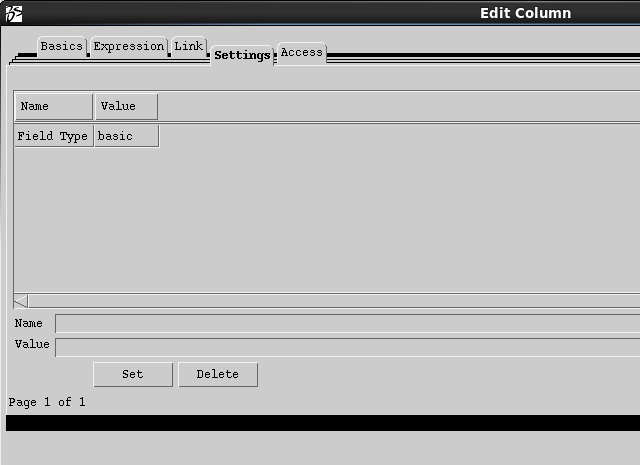
名称



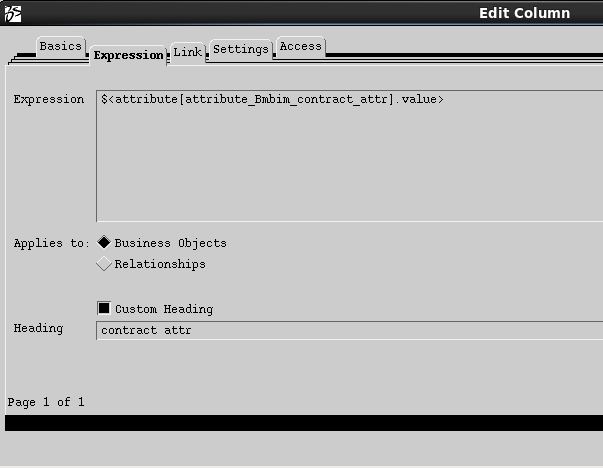


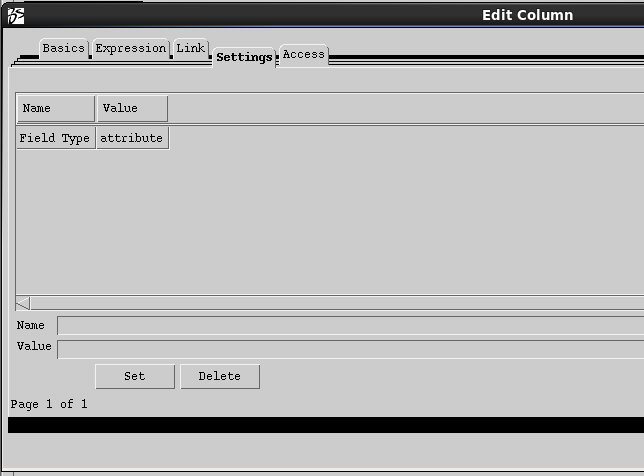
类型



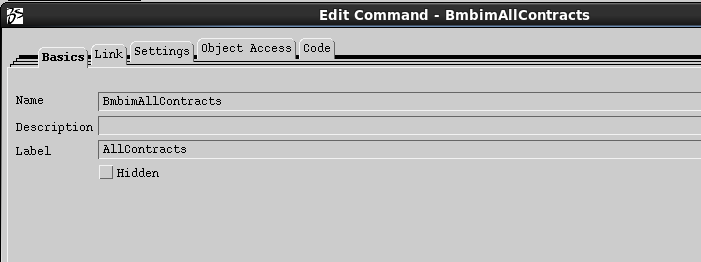


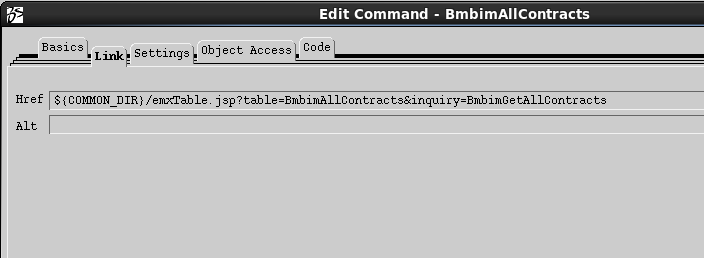
合同属性





1. 创建一个command(BmbimAllContracts),关联inquiry(BmbimGetAllContracts)和table(BmbimAllContracts)

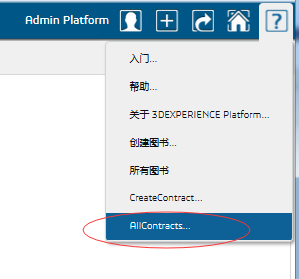


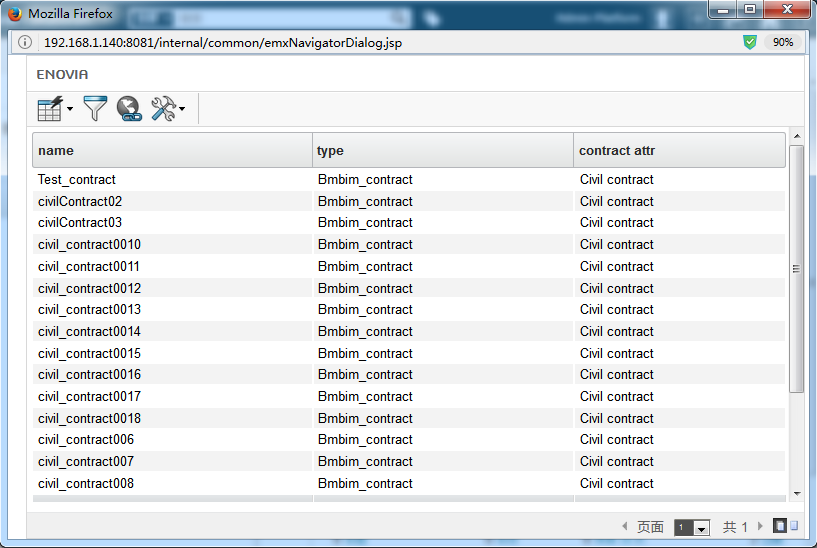


在系统中关联此command(BmbimAllContracts)



1. 运行系统，查看结果

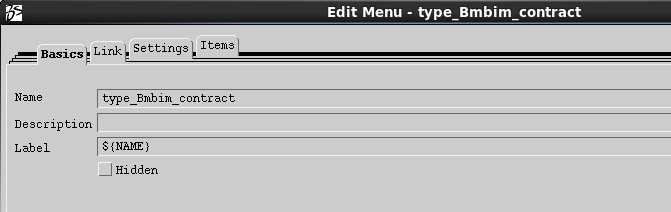


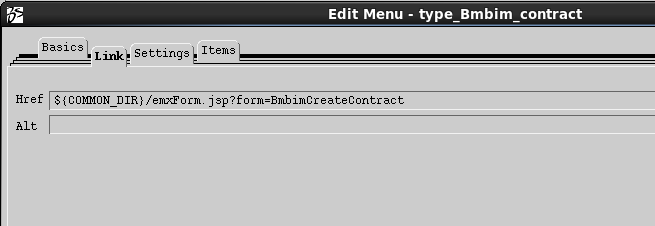


5.配置一棵树(type\_Bmbim\_contract)，展示单个合同的信息,并进行修改

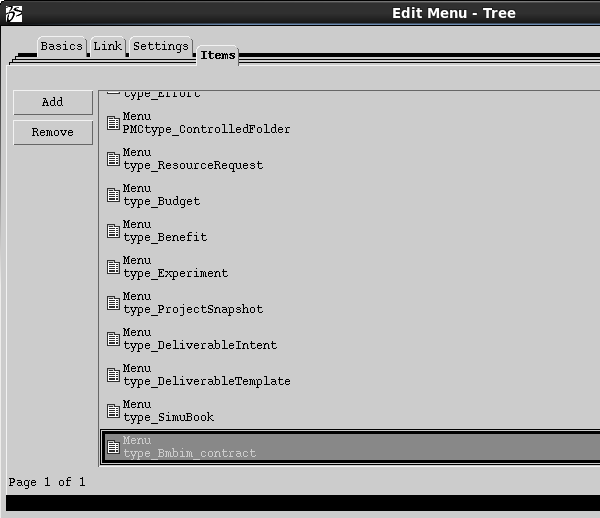
注意：树也就是menu，通常以type\_开头，后面加上type的注册时的原始名称

1. 创建一个menu(type\_Bmbim\_contract)

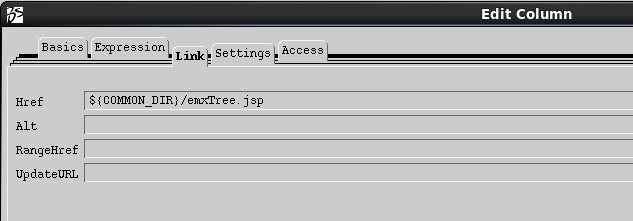




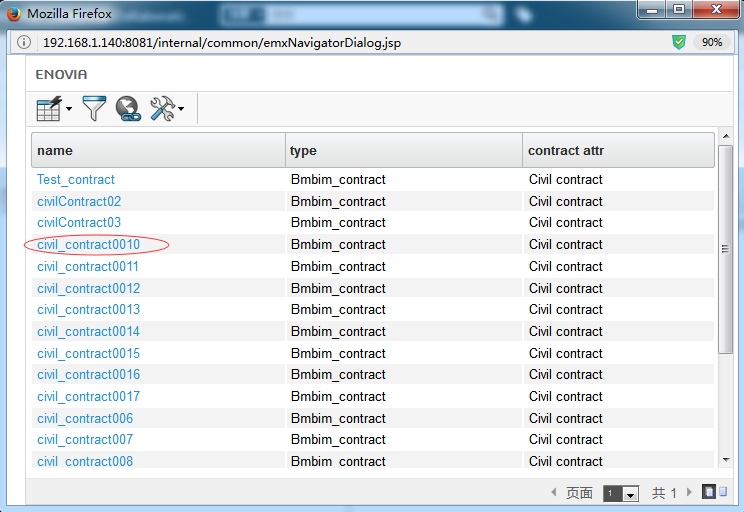
1. 将menu(type\_Bmbim\_contract)关联到menu(Tree)中

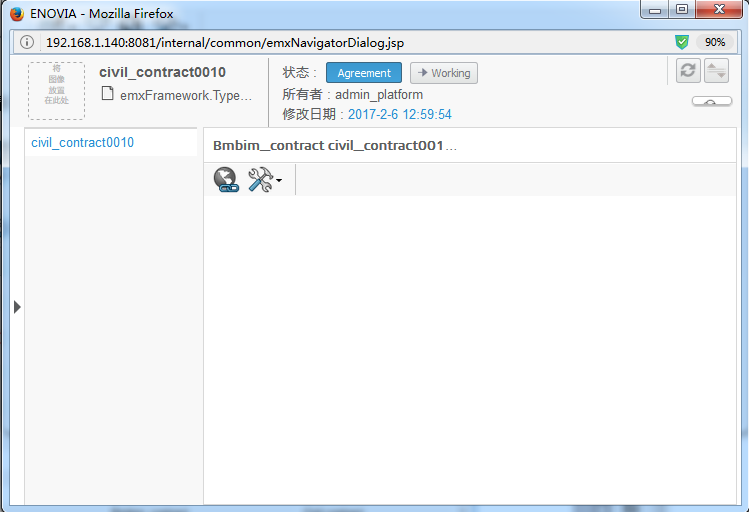


1. 修改table(BmbimAllContracts)的列name



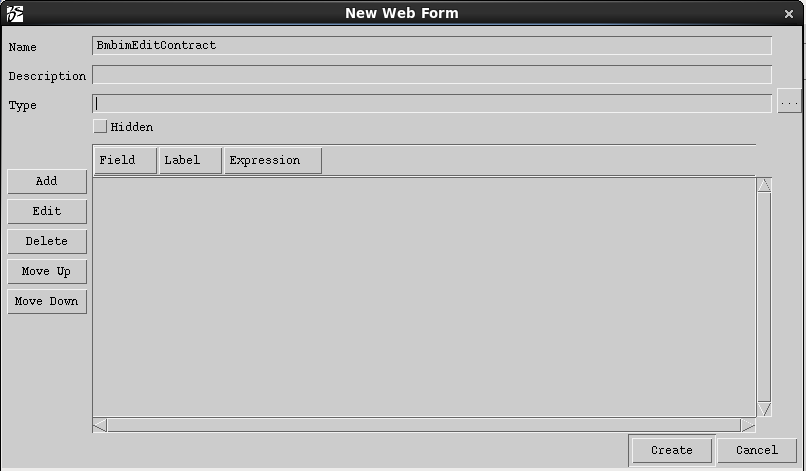
1. 在系统中查看效果





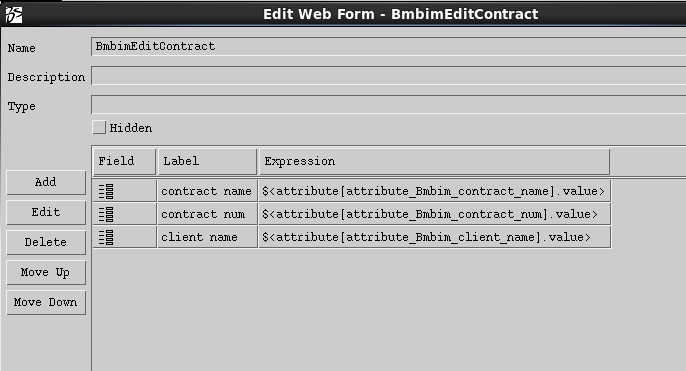
1. 创建一个修改合同的页面

（1）创建一个webform(BmbimEditContract)



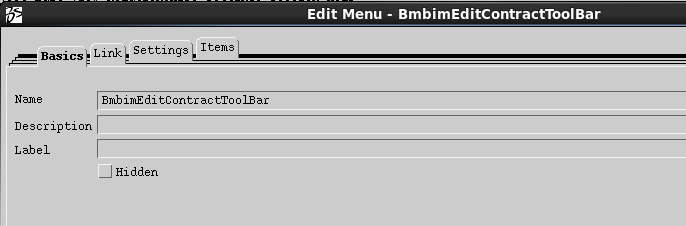
添加基本列

参考上面webform(BmbimCreateContract)中的添加字段部分的内容



（2）配置一个工具栏，也就是menu,

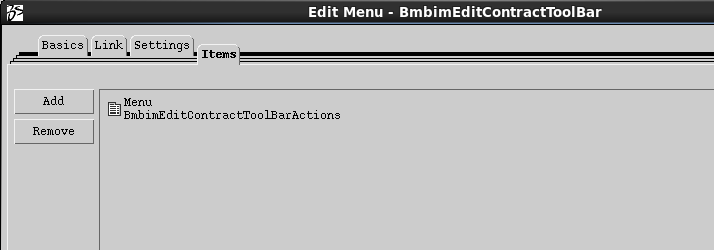
menu(BmbimEditContractToolBar)



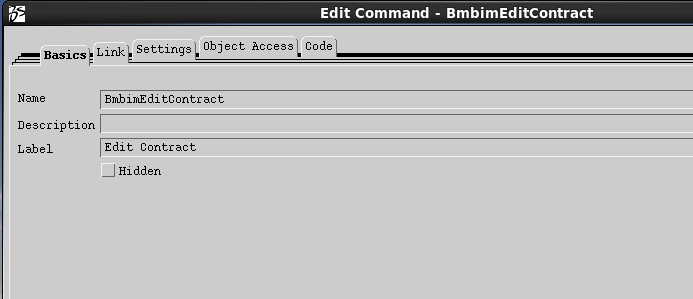
menu(BmbimEditContractToolBarActions)

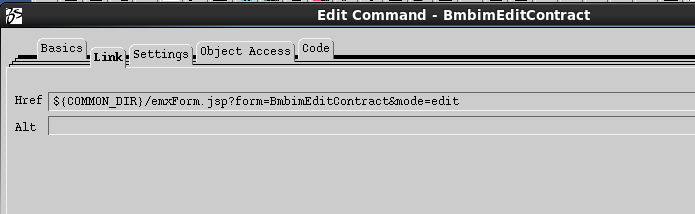


关联上面两个menu

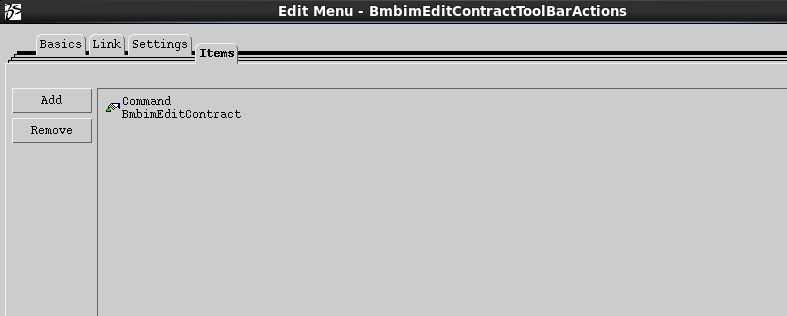


1. 创建一个command(BmbimEditContract)

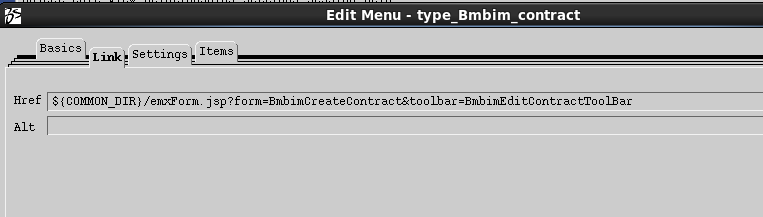




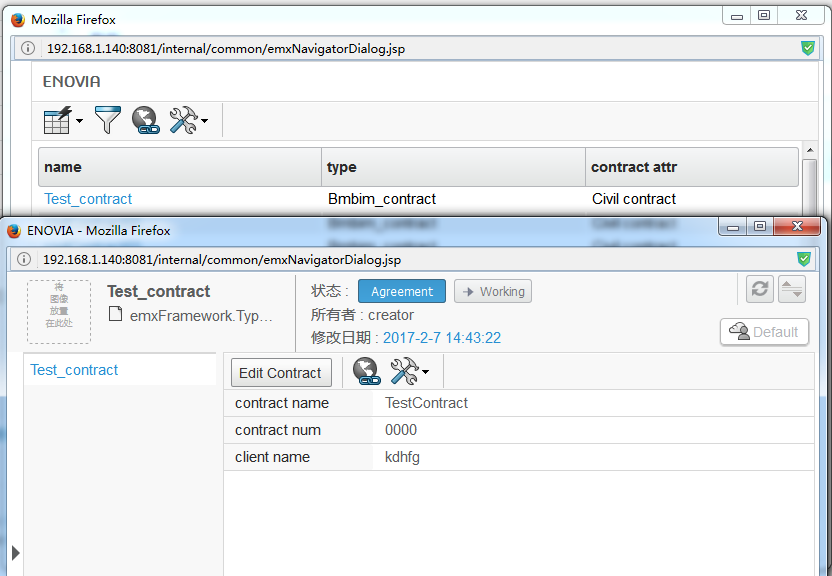
将command(BmbimEditContract)关联到menu(BmbimEditContractToolBarActions)



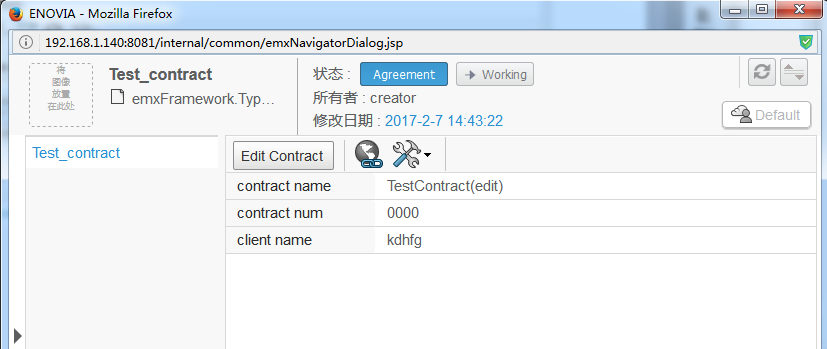
1. 最后在树（type\_Bmbim\_contract）的link中加入工具栏menu(BmbimEditContractToolBar)



1. 运行效果



修改成功：



小结：inquiry通过mql从数据库中查询数据，table展示多个合同对象的信息，对象树的配置展示单个合同的信息。

# 删除合同对象

1. 创建一张JSP实现删除

在工程文件/home/webapp/data/as16x01/as-3dspaceNoCas/webapps/internal/common

下创建BmbimDeleteContracts.jsp

Jsp代码内容为实现删除对象，其中需引用系统inc文件和DomaiObject类

代码如下：

<%--

static const char RCSID[] = "$Id: BmbimDeleteContracts.jsp,v 1.0 2017/02/07 15:01:02 mengzq Exp $";

--%>

<%@include file="../emxUICommonHeaderBeginInclude.inc"%>

<%@include file="../emxUICommonAppInclude.inc"%>

<%@page import="com.matrixone.apps.domain.DomainObject"%>

<%

String[] contractsID = emxGetParameterValues(request, "emxTableRowId");

try{

DomainObject.deleteObjects(context, contractsID);

}catch(Exception e){

e.printStackTrace();

}

%>

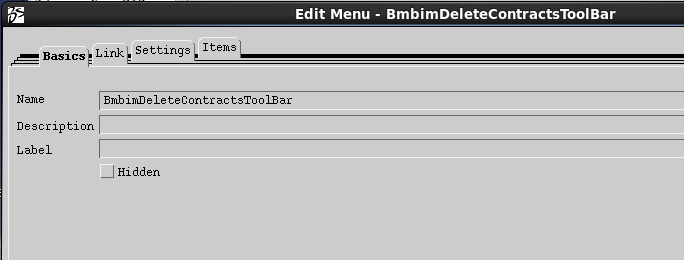
<script language="javascript" type="text/javaScript">

top.refreshTablePage();

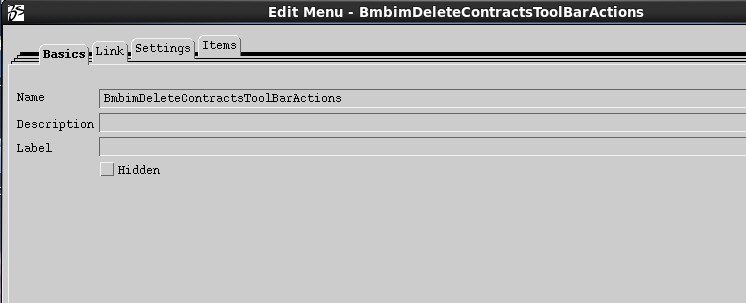
</script>

1. 创建一个工具栏

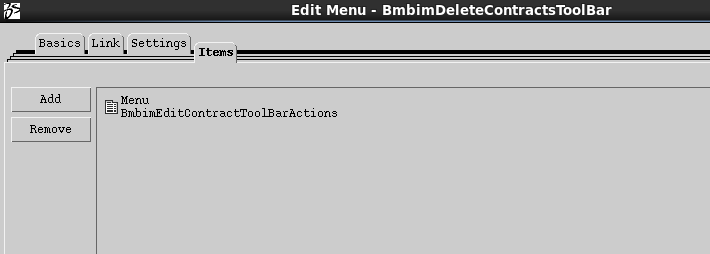
menu(BmbimDeleteContractsToolBar)



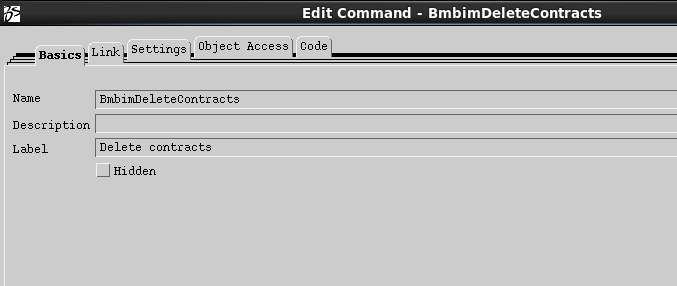
menu(BmbimDeleteContractsToolBarActions)

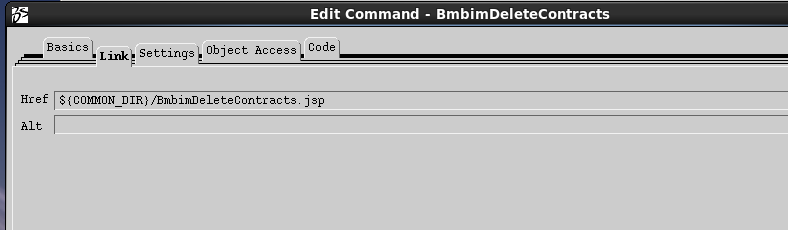


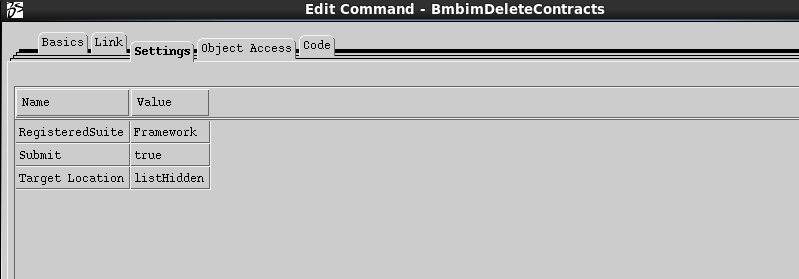
关联上面两个menu



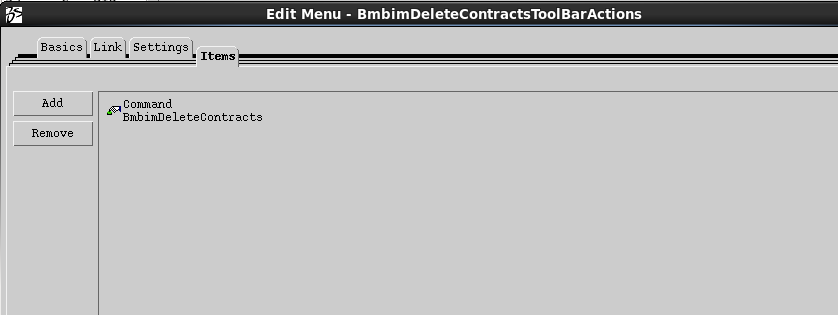
1. 创建一个command(BmbimDeleteContracts)



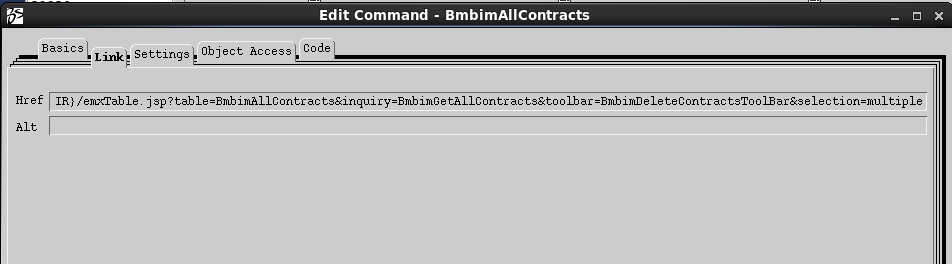




将command关联到menu(BmbimDeleteContractsToolBarActions)

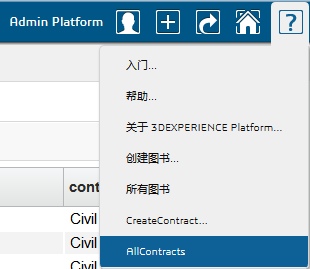


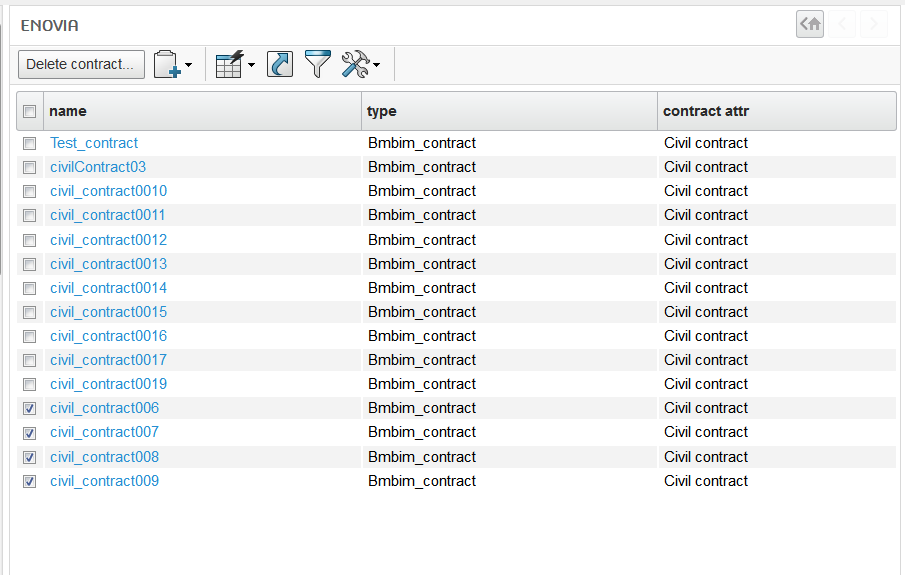
1. 在系统中关联menu(BmbimDeleteContractsToolBar)

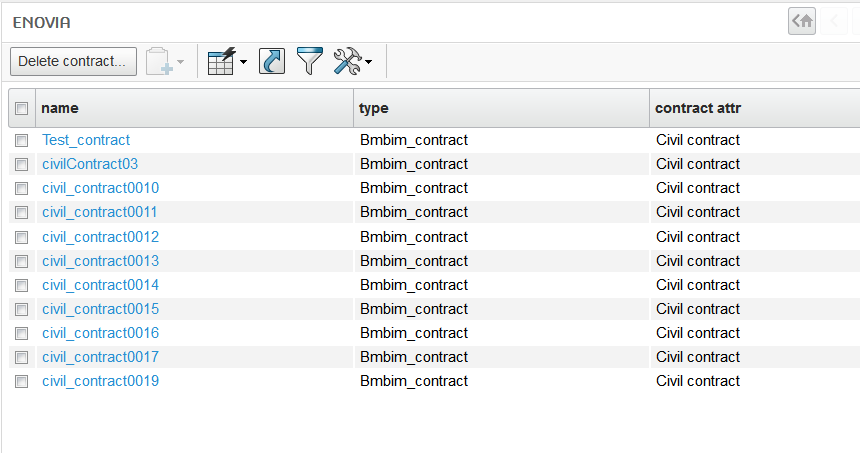




1. 运行效果







小结：删除功能的实现：

1. 通过组件command(BmbimDeleteContracts)配置参数（link=${COMMON\_DIR}/BmbimDeleteContracts.jsp）,实现入口去调用JSP;
2. BmbimDeleteContracts.jsp中通过请求，操作数据库删除合同对象，用到一个最基本并且核心的类：DomainObject类，是所有BO对象的基类。

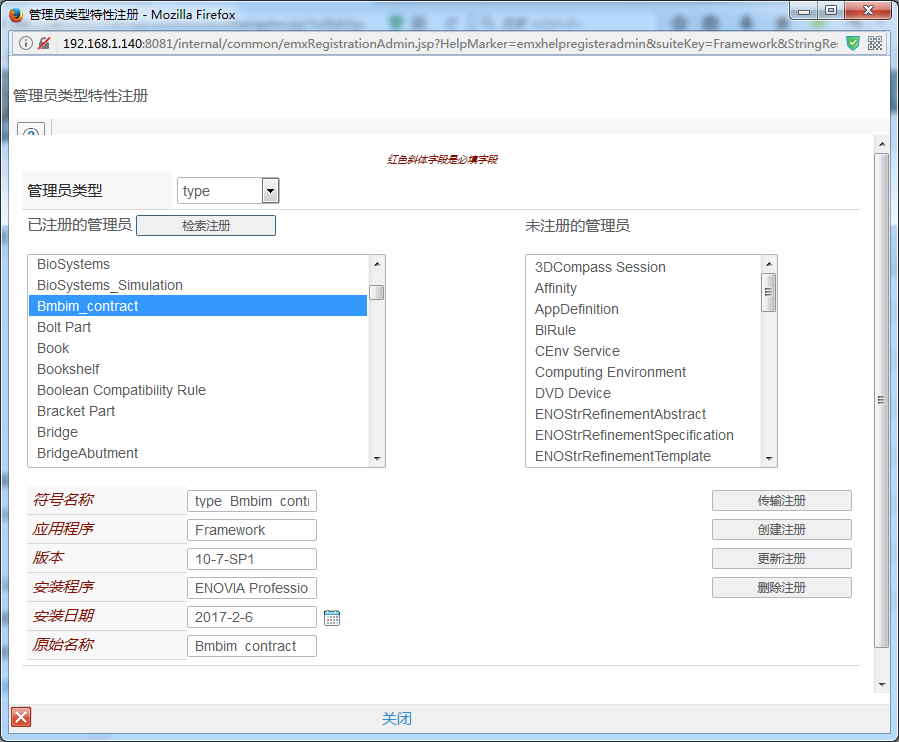
# 注册

1. 注册机制

注册机制其实就是别名管理，严格说来，任意一个AO都需在系统中注册，注册的好处就是，假如需求改动时需改动变化一个AO的名字，而功能模块已做好，这时直接改Business中的AO名字的话，需变动很多代码，维护和扩充难度很大。这时只改动AO的注册名，则需求一样实现，并且不用改动代码块。

1. 注册说明





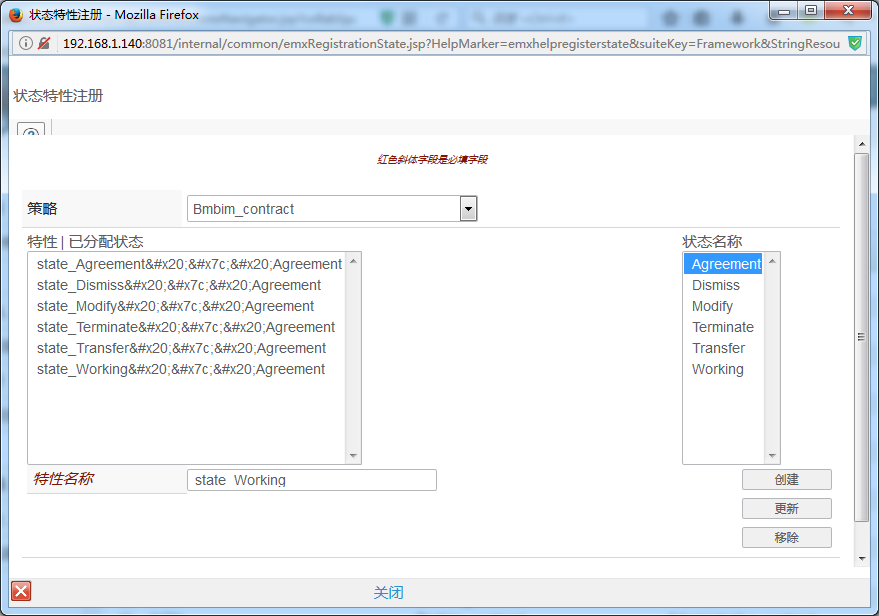
在管理类型中选择相应的类型即可

需注意的是“符号名称”必须以相应的类型的小写字母+\_开头，如type\_Bmbim\_contract，其中“Bmbim\_contact”自定义，不过建议按规范书写，即此type在Business中的名字

更改时，点击刷新注册即可，系统自动重载生效

3.Policy的state的注册：





备注：policy的state必须注册才能使用

# 国际化

系统的国际化与java的国际化相同，需注意的是一些固定格式

1scheam的国际化写入emxFrameworkStringResource\_zh.properties文件里

2其它国际化，如command\menu\webform.etc.的写入相应的注册模块里，

3Scheam国际化的具体写法

（1）emxFramework.具体类型.AO名(有空格用\_代替)，适用于type,attr,rel,policy,state

如合同的 emxFramework.Type.Bmbim\_contract

（2）特殊的Attr的range值 emxFramework.Range.attr名(有空格用\_代替).range值名(有空格用\_代替) 如合同属性的港口 emxFramework.Range.Bmbim\_contract\_attr.Port

Attr的默认值：emxFramework.Default.attr名(有空格用\_代替)

如合同类别的默认值：emxFramework.Default.Bmbim\_contract\_attr

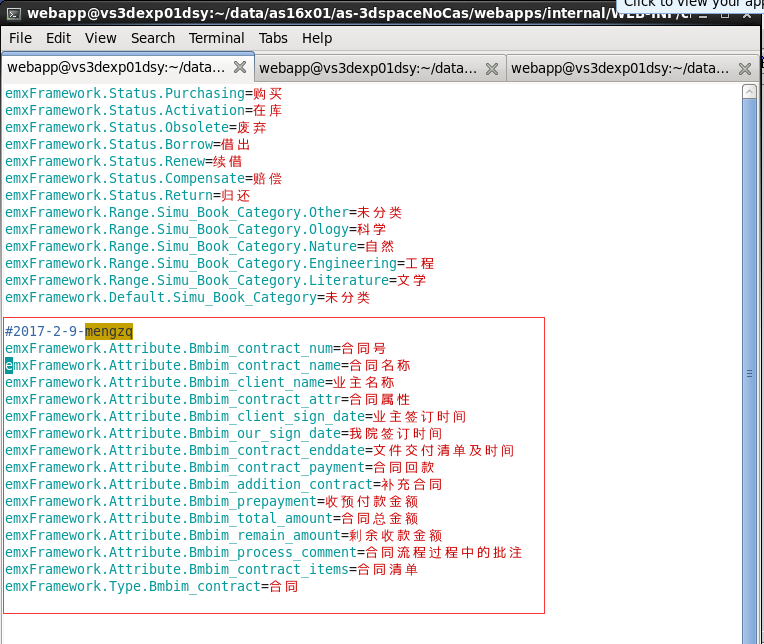
1. policy的state

emxFramework.State.policy名.state名

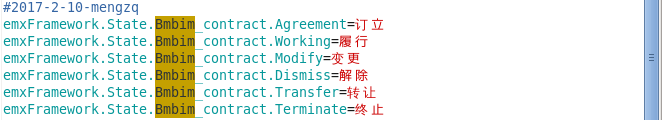
1.找到以下目录：(ToMEE下的internal文件夹在编译后，修改的属性文件内容会失效，最好是修改3dspace安装目录下的internal的属性文件，然后进行重新编译)

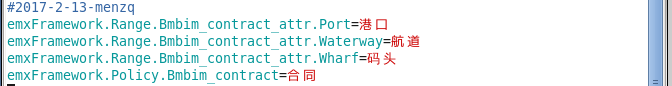
/home/webapp/data/as16x01/as-3dspaceNoCas/webapps/internal/WEB-INF/classes

在emxFrameworkStringResource\_zh\_CNNative.properties中添加内容



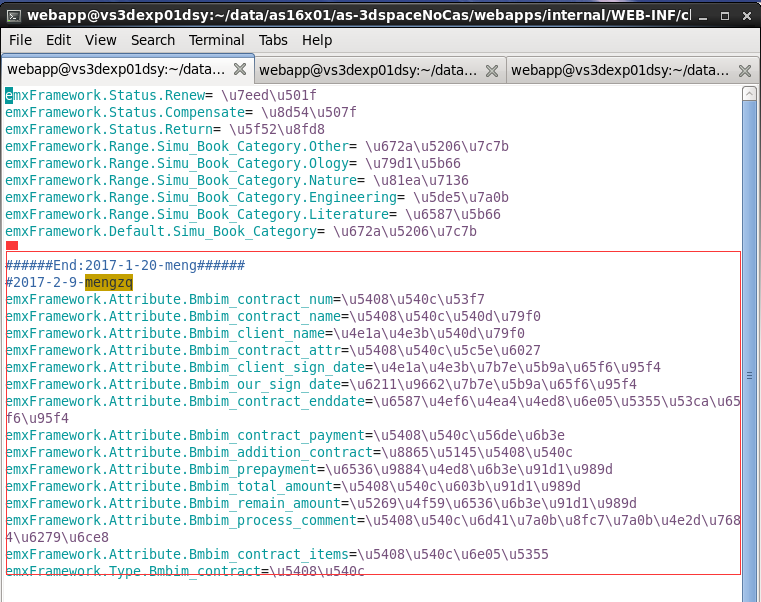


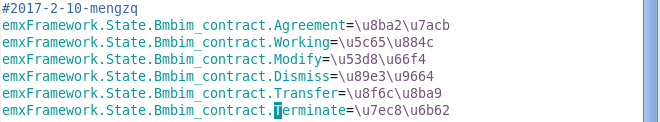


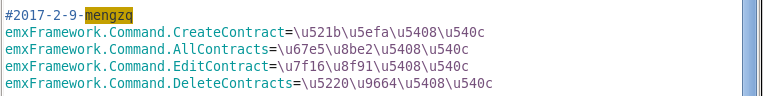


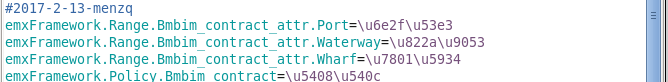
1. 在emxFrameworkStringResource\_zh\_CN.properties中添加内容

(在线编码转换;http://tool.oschina.net/encode?type=3)



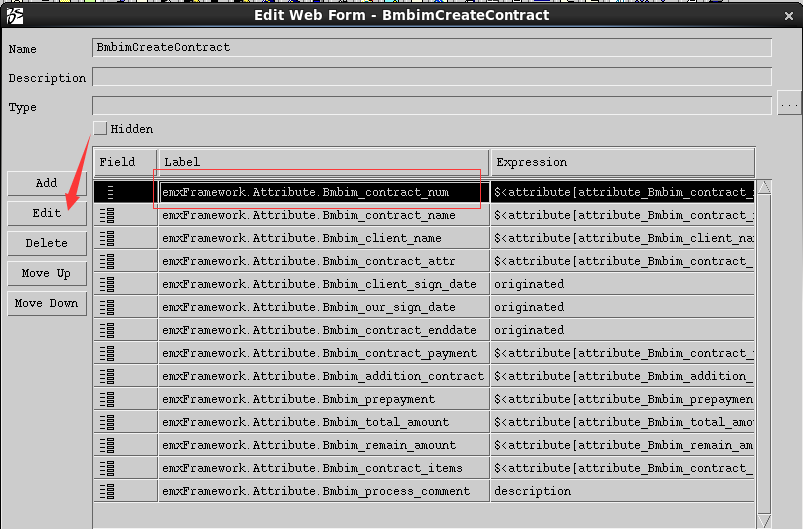


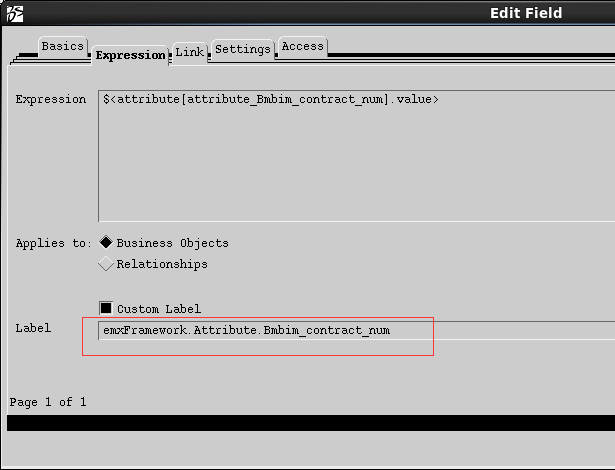




1. 添加参数

找到需要国际化对象的label参数





Label必须和属性文件添加内容的等号左边的值要一致，书写规则：

emxFramework.类型.注册名

settings添加以下参数值：

