

Teamcenter定制开发课程

Teamcenter custom development courses

第八课：服务端Internal ITK开发之 Business Modeler Framework (BMF)



作者：王江江
邮件：James.wang@vdior.com

二零一一年五月|上海

课程目标

- 知道BMF开发原理；
- 掌握BMF开发方式；



BMF开发原理

■ 一些概念描述

- Method 是使用者可以定义的，跟类型（type）及属性（property）动作相关的功能
- Actions 动作：譬如新建、删除、保存等等，一个动作的完成依序为
 - ✓ Pre-condition
 - ✓ Pre-action
 - ✓ Base action
 - ✓ Post-action
- 搜寻在 %TC_INCLUDE% 目录内文件名为 *_msg.h，即可知系统提供了哪些 message，可以定义 pre-condition/pre-action/post-action
- Message 一样有继承的关系，譬如 WorkspaceObject 的 message，则 item, form, folder 等等均可使用。同样的 message 如果都存在，优先调用该 type/class 的 message

BMF开发原理

➤ 使用 BMF 的方式

1. 设置 Site Preference : **BMF_CUSTOM_IMPLEMENTOR_PATH**

- ✓ 此 Preference 允许 multiple values , 可以设置多个路径
- ✓ 必须设定绝对路径 , 不可使用相对路径

2. 于 BMIDE 设置对应的 extension rule

3. 编写 extension function

```
extern USER_EXT_DLL_API int your-function-name  
                                (METHOD_message_t*, va_list);
```

4. 将 extension rule 定义的 library 文件复制到 BMF_CUSTOM_IMPLEMENTOR_PATH 指定的任一目录中

BMF开发样例

◆开发样例

一. 做一个PostAction，当Item创建完成后，设置版本Form描述字段为某一固定值。

1. 新建工程项目libZR2imf(ZR2是我IDE项目中的前缀，根据IDE项目不同而前缀不同)。
2. 编写设置属性函数ZR2Item_set_master_form_value(METHOD_message_t *msg, va_list args)，

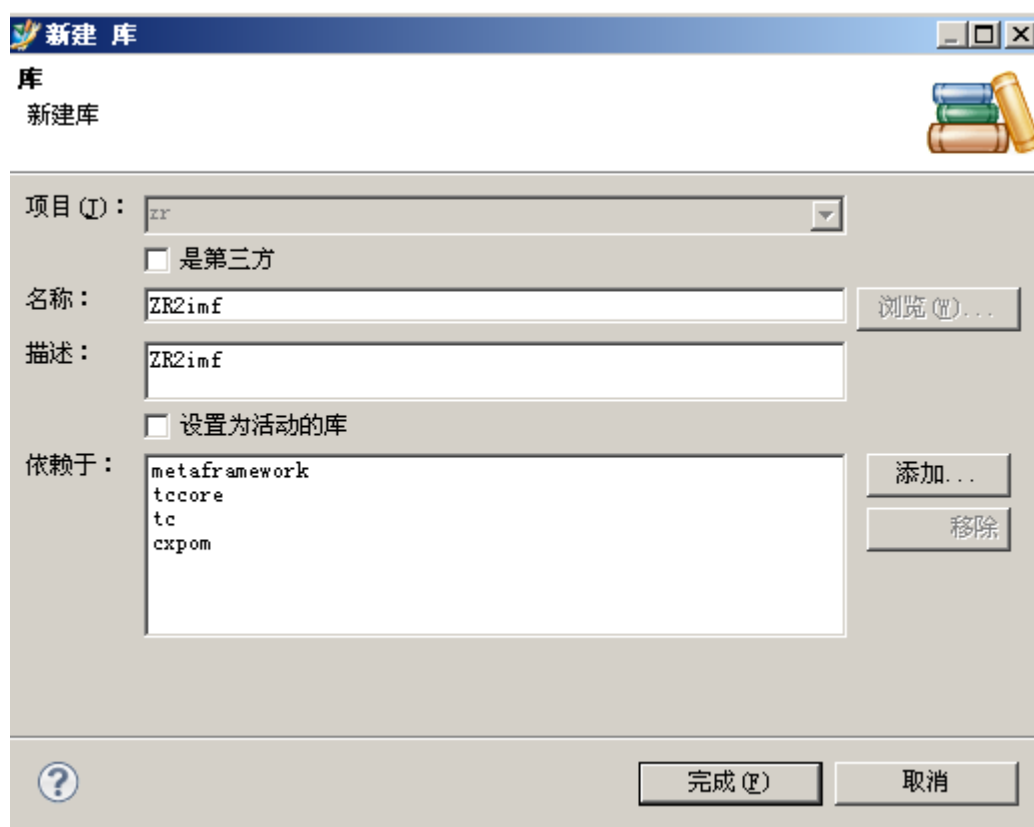
代码如下图所示:

```
int ZR2Item_set_master_form_value(METHOD_message_t *msg, va_list args)
{
    int ifail = ITK_ok;
    char buf[BUFSIZ] = "";
    tag_t template_ds = NULLTAG;
    char *item_id = va_arg( args, char* );
    char *item_name = va_arg( args, char* );
    char *type_name = va_arg( args, char* );
    char *rev_id = va_arg( args, char* );
    tag_t *new_item = va_arg( args, tag_t* );
    tag_t *new_rev = va_arg( args, tag_t* );
    tag_t msg_item = *new_item, msg_rev = *new_rev;
    tag_t master_form_rel_type = NULLTAG;
    ITKCALL( GRM_find_relation_type( "IMAN_master_form", &master_form_rel_type ));
    int form_count = 0;
    tag_t *form_list = NULL, master_form = NULLTAG;
    ITKCALL( GRM_list_secondary_objects_only( msg_rev, master_form_rel_type, &form_count, &form_list ));
    if( form_count > 0 )
    {
        master_form = form_list[0];
        ITKCALL( AOM_lock( master_form ));
        ITKCALL( AOM_set_value_string( master_form, "object_desc", "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!" ));
        ITKCALL( AOM_save( master_form ));
        ITKCALL( AOM_unlock( master_form ));
    }
    MEM_free( form_list );
    printf( "Begin ITEM_create_msg method \n" );
    return ITK_ok;
}
```

BMF开发样例

3. 编译代码，并生成libZR2imf.lib文件，放到某一目录。

4. 打开IDE，在代码生成库中新建库ZR2imf，如下图所示：



BMF开发样例

5. 在规则，扩展规则中，新建扩展ZR2Item_set_master_form_value，如下图所示：

新建 扩展定义...

扩展
新建扩展

项目 (I):

名称 (A):

语言 (G):

库 (L):

☐ 内部的?

参数列表:

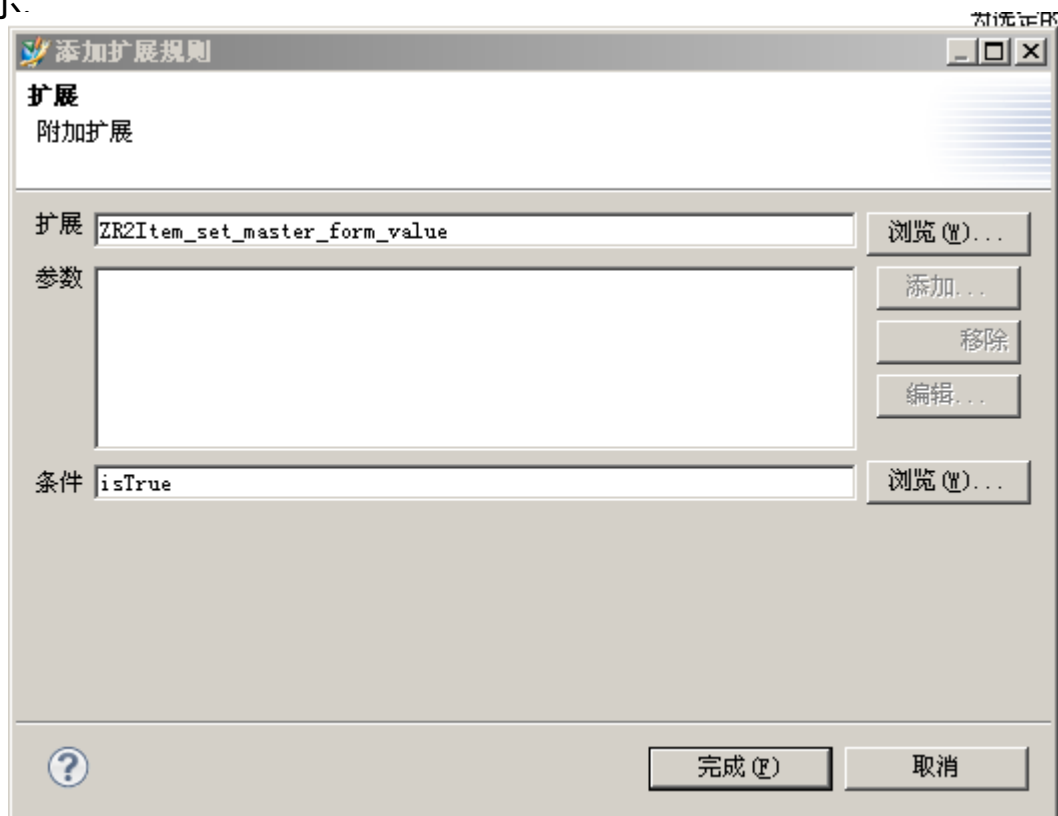
名称	类型	必需的	建议值	LOV/查询名称

可用性:

业务对象	属性	属性名称	操作名称	扩展点
Item			ITEM_create	PostAction

BMF开发样例

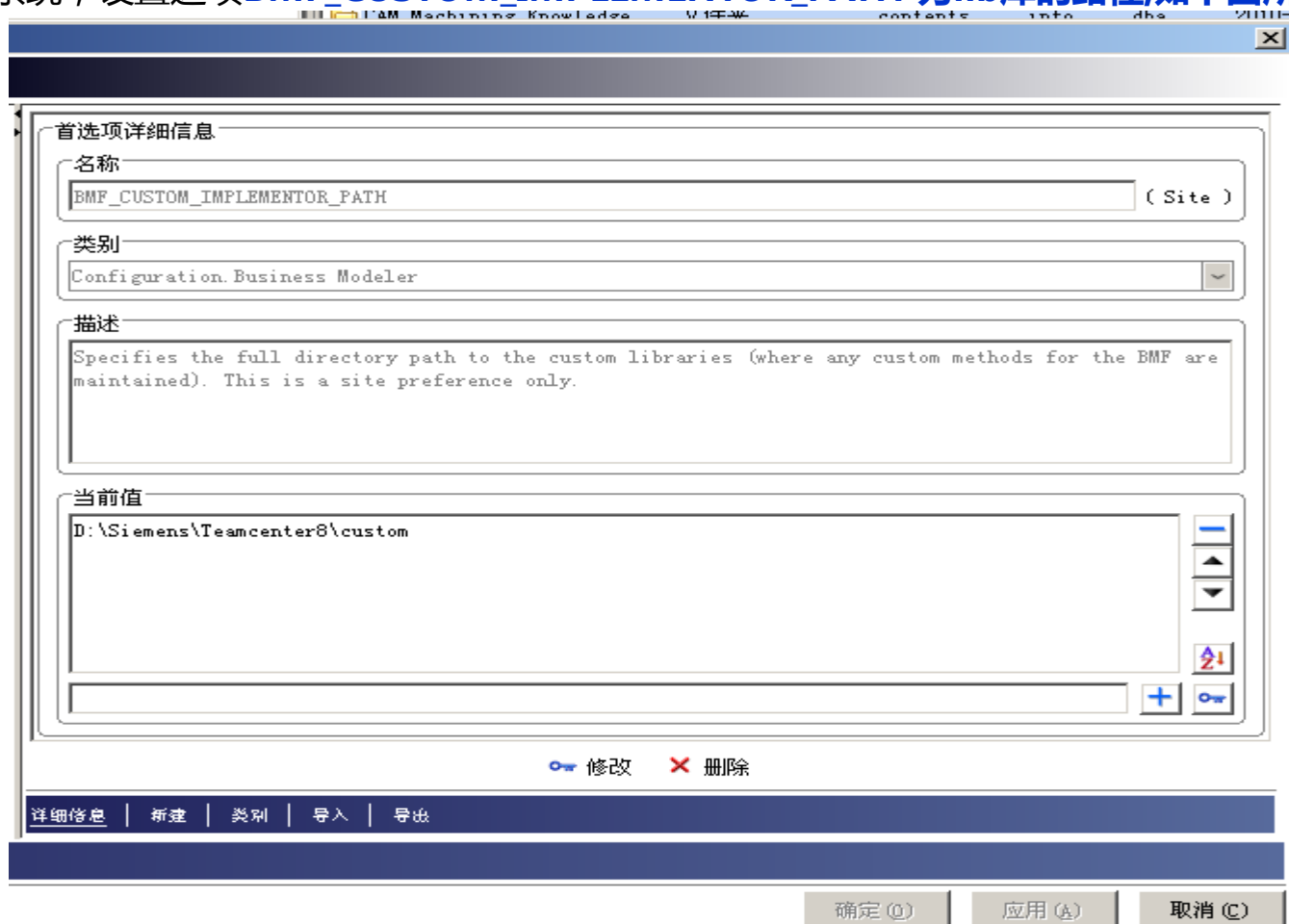
6. 选择要赋予的业务对象Item，并选择操作，原生操作，制定到ITEM_create,选择添加post Action,如下图所示:



BMF开发样例

7. 保存并部署到系统。

8.进入系统，设置选项**BMF_CUSTOM_IMPLEMENTOR_PATH** 为lib库的路径,如下图所示:



The screenshot shows a configuration window titled "BMF Custom Implementor Path". It contains the following fields and controls:

- 名称 (Name):** BMF_CUSTOM_IMPLEMENTOR_PATH (Site)
- 类别 (Category):** Configuration Business Modeler
- 描述 (Description):** Specifies the full directory path to the custom libraries (where any custom methods for the BMF are maintained). This is a site preference only.
- 当前值 (Current Value):** D:\Siemens\Teamcenter8\custom
- 操作按钮:** 修改 (Modify), 删除 (Delete)
- 底部菜单:** 详细信息 (Details), 新建 (New), 类别 (Category), 导入 (Import), 导出 (Export)
- 底部按钮:** 确定 (OK), 应用 (Apply), 取消 (Cancel)

BMF开发样例

9. 重启客户端，然后进行新建Item测试。

二. 添加新建Item时的规则验证Pre-condition，当ItemID小于8个字符的时候提示错误信息。

1. 编写函数int ZR2ValiteId(METHOD_message_t *msg,va_list args)，定义Item ID验证的处理逻辑代码，代码如下所示:

```
int ZR2ValiteId(METHOD_message_t *msg,va_list args)
{
    int ifail = ITK_ok;
    char buf[BUFSIZ] = "";
    tag_t template_ds = NULLTAG;
    char *item_id = va_arg( args, char*);
    printf( "item_id= %s", item_id);
    if(strlen(item_id)>8)
    {
        return ifail;
    }
    else
    {
        return 919000+1568;
    }
}
```

BMF开发样例

2. 在IDE中新建扩展规则ZR2ValiteId，并赋予扩展规则到业务对象Item的ITEM_create操作的PreCondition。如下图所示:

新建 扩展定义...

扩展
新建扩展

项目 (I):

名称 (A):

语言 (G):

库 (L):

☐ 内部的?

浏览 (B)...

参数列表:

名称	类型	必需的	建议值	LOV/查询名称

添加...
编辑...
移除
上移 (U)
下移 (D)

可用性:

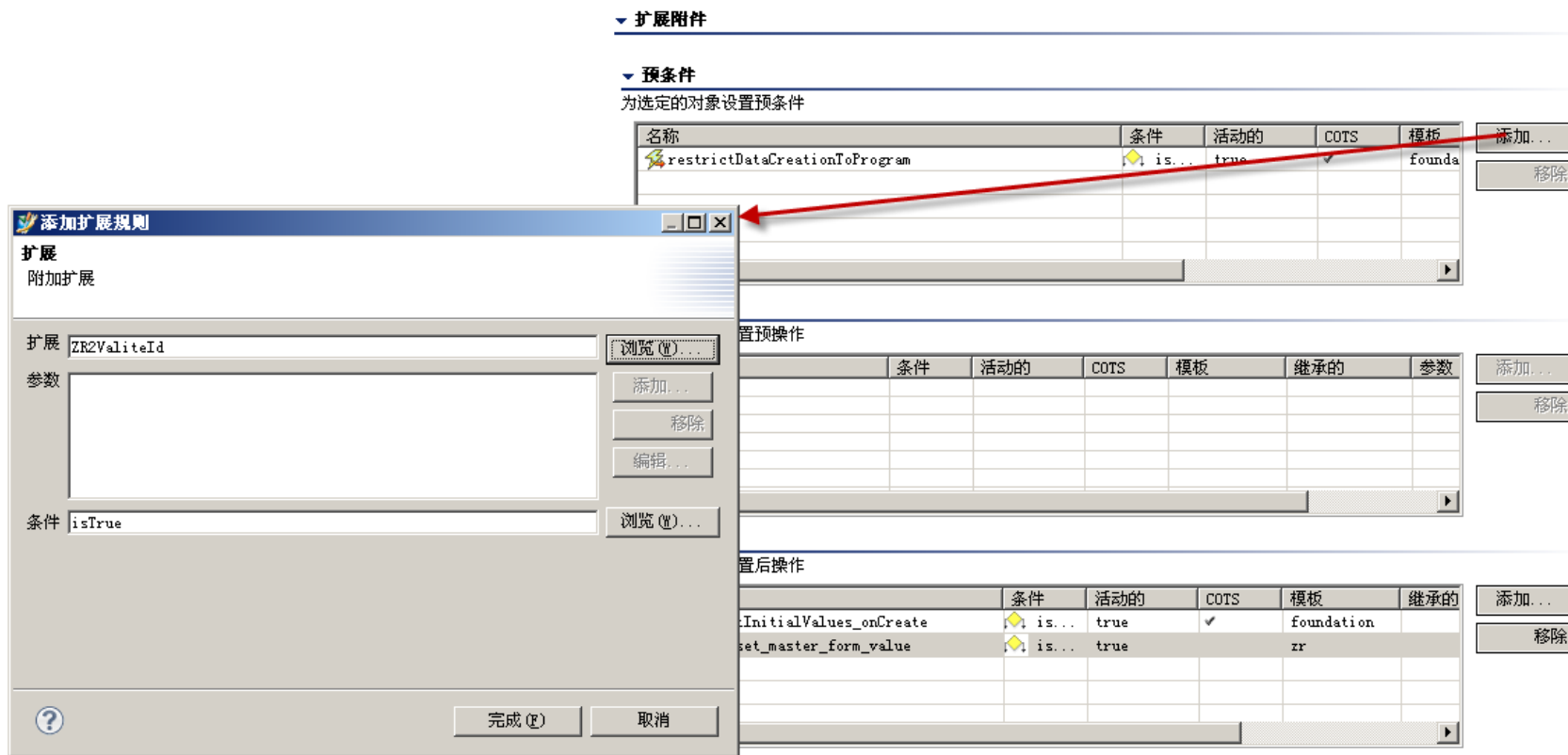
业务对象	属性	属性名称	操作名称	扩展点
Item			ITEM_create	PreCondition

添加...
编辑...
移除

完成 (F) 取消

BMF开发样例

3. 在Item的原生操作ITEM_create中进行PreCondition添加，如下图所示:



BMF开发样例

4. 定义报错信息

系统允许定义的错误编号

- 使用者自订错误编号范围为 919001~919999
- 常数 EMH_USER_error_base 919000 定义在 %TC_INCLUDE%\tc\ emh_const.h
- 定义错误讯息步骤
 1. 新增头文件 [cust_errors.h](#)
 2. 于 cust_errors.h 内定义错误编号，语法

```
#define USER_ERROR_NAME (EMH_USER_error_base + number)
```
 3. 进入TCROOT\lang\textserver\zh_CN,在ue_errors.xml 内加上错误编号代表的信息，如

```
<error id= "number" > [提示信息]</error>
```

 - ✓ *number* 与 cust_errors.h 中定义的编号相同

定义ID校验的错误提示信息，进入TCROOT\lang\textserver\zh_CN目录，代码如下：

```
<textsrv filename="ue_errors.xml">
<errors module="ue" error_base="919000">
<error id="1568">编码ID不符合长度要求</error>
</errors>
</textsrv>
```

BMF开发样例

4. 重启Teamcenter系统进行创建Item的测试.效果如下：

课后练习

1. POST Action 在Item创建完成后，在Item Revision下自动添加两个数据集。
2. Pre-action Action，在Item创建时，验证Item编码中是否含有Y，不然不允许创建。

谢谢，有什么问题吗？

下一课：