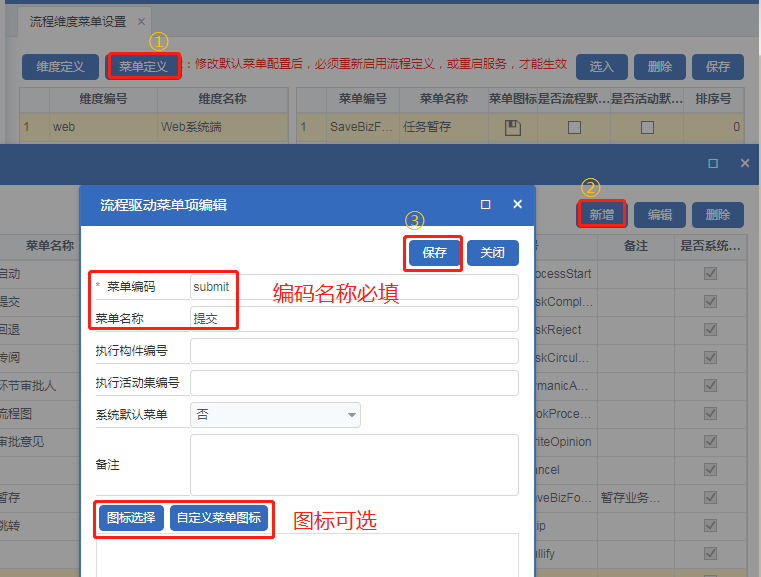
Vbase设置

第一步：在流程管理->基础设置->流程维度菜单设置页面新增一个菜单项，其中菜单编码和菜单项必填，其它可不填，菜单编码用来唯一区分菜单编码的，菜单名称是用来显示的。如：新增编码submit和名称为自定义提交的菜单项。

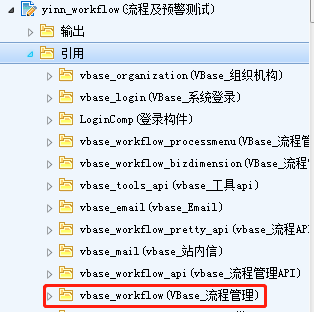


第二步，把刚刚新增的菜单项选入并保存，这样vbase流程框架才能看到这个菜单。其中，是否流程默认菜单是指流程启动时该菜单是否默认显示，是否活动默认菜单是指审批任务时该菜单是否默认显示，排序号是指显示的顺序。如，这里勾选了活动默认菜单。

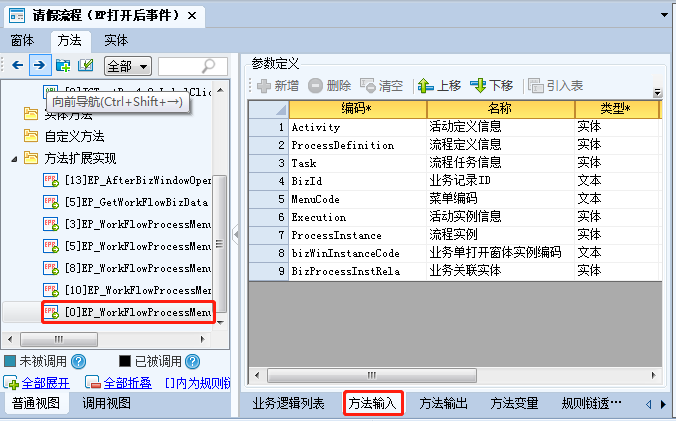


开发平台配置

第三步，构件引用vbase\_workflow



第四步，配置方法扩展实现EP\_WorkFlowProcessMenuExecute，并输出，可在构件方法中配置，但是建议在打开的业务审批表单窗体处配置。其中这些方法输入是已知数值的变量，可直接使用。



接着开始实现自己的逻辑，这里实现的场景是，提交时不使用vbase的下一步审批人公共窗体，而是自定义窗体实现选择活动节点、活动审批人和填写审批意见的功能。

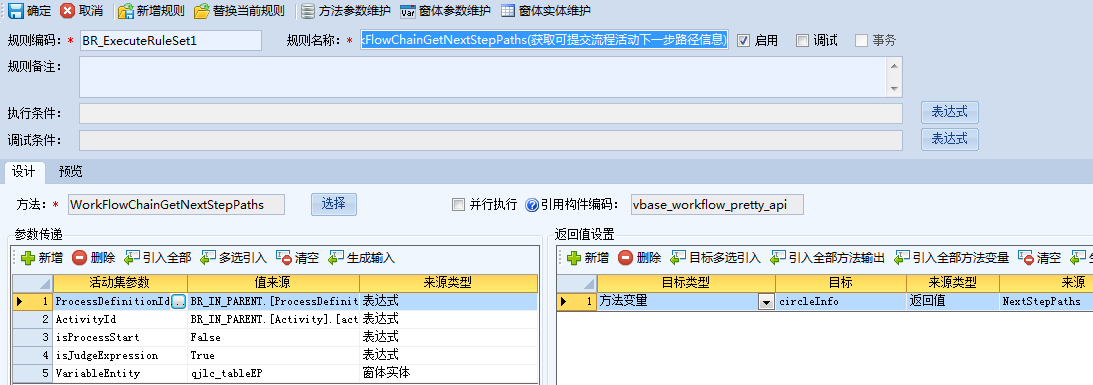
整体规则链配置如下：



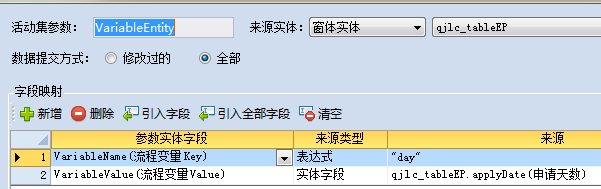
（1）判断编码是否和刚才新增的编码一致，一致才往下执行，不一致就不执行，因为只要一点维度菜单所有的扩展点都会执行，根据判断语句来控制有效执行规则链。

判断条件为：BR\_IN\_PARENT.MenuCode=="submit"

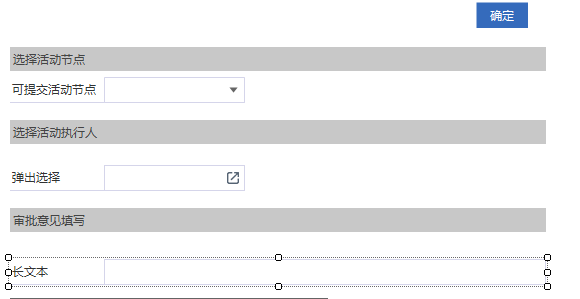
（2）执行方法vbase\_workflow\_pretty\_api. WorkFlowChainGetNextStepPaths(获取可提交流程活动下一步路径信息)，获取可以提交的活动信息。



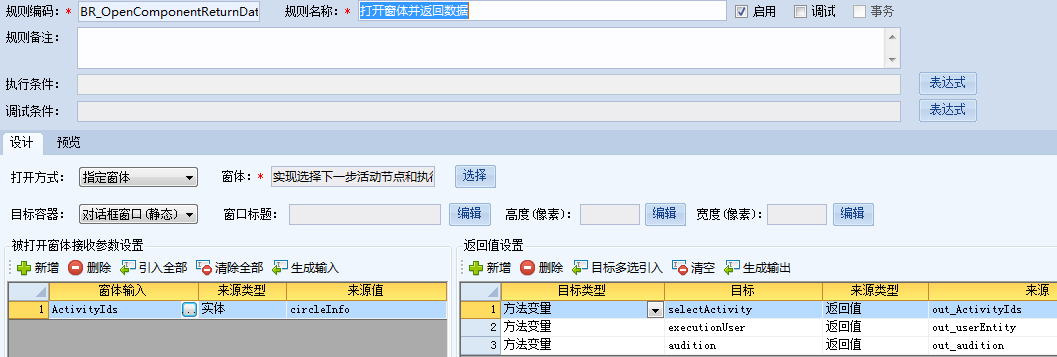
自己做方法变量或者窗体实体来接收返回值，这里用的是方法变量。其中，ProcessDefinitionId传入值BR\_IN\_PARENT.[ProcessDefinition].[processDefinitionId]，ActivityId传入值BR\_IN\_PARENT.[Activity].[activityId]，VariableEntity流程变量实体根据流程定义确定是否需要传，这里传的是如下图



（3）打开窗体并返回数据，这里打开的是自己做的一个选择执行人和活动节点的窗体。如下图



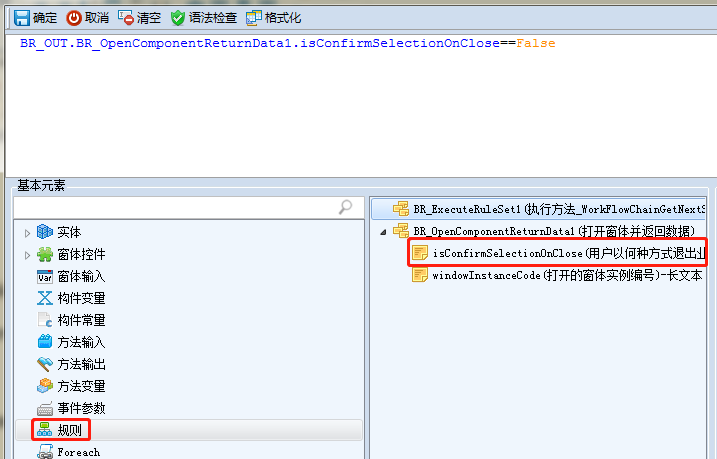
将刚刚可选择活动节点传入窗体输入，并用方法变量接收窗体输出，这里几乎是赋值操作，不做赘述。



（4）再次判断是正常退出窗体还是点X直接关闭静态窗体，判断条件为：

BR\_OUT.BR\_OpenComponentReturnData1.isConfirmSelectionOnClose==False

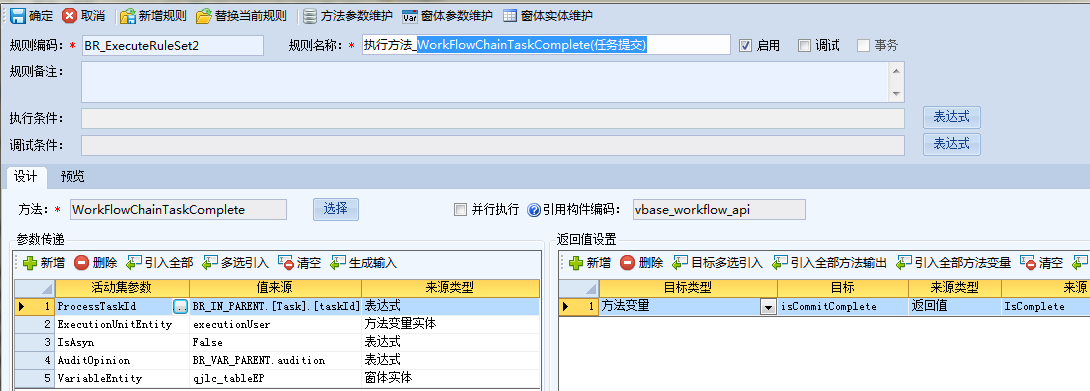
意思是如果不是正常退出窗体，就中断后续操作，免得引起不必要的报错和冗余数据。



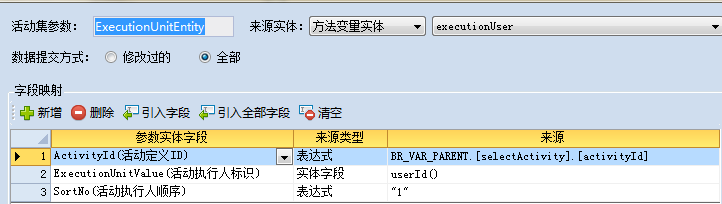
（5）中断规则。

（6）调试用的提示信息，可忽略。

（7）执行方法vbase\_workflow\_api. WorkFlowChainTaskComplete(任务提交)。



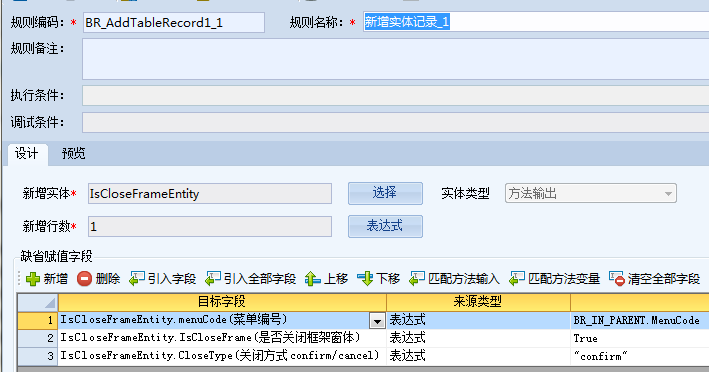
其中，processTaskId传入值为BR\_IN\_PARENT.[Task].[taskId]，ExecutionUnitEntity审批人实体传入值如下，这是必填项确定活动ID和执行人的实体，这两个值都是刚刚用方法变量接收的窗体输出的值。



方法变量实体可选，是否异步执行可选，审批意见可选（也是接收的窗体输出值）。

（8）判断任务提交是否成功。

（9）关闭vbase审批框架。



窗体demo在yinn\_workflow构件，流程表单目录，form23(请假流程EP打开后事件)窗体的EP实现中