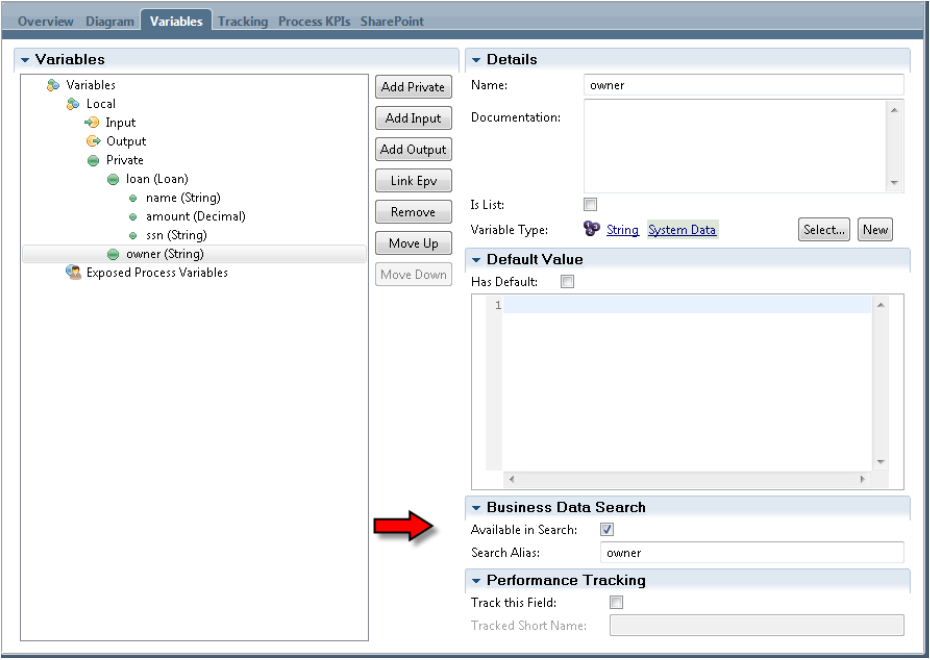
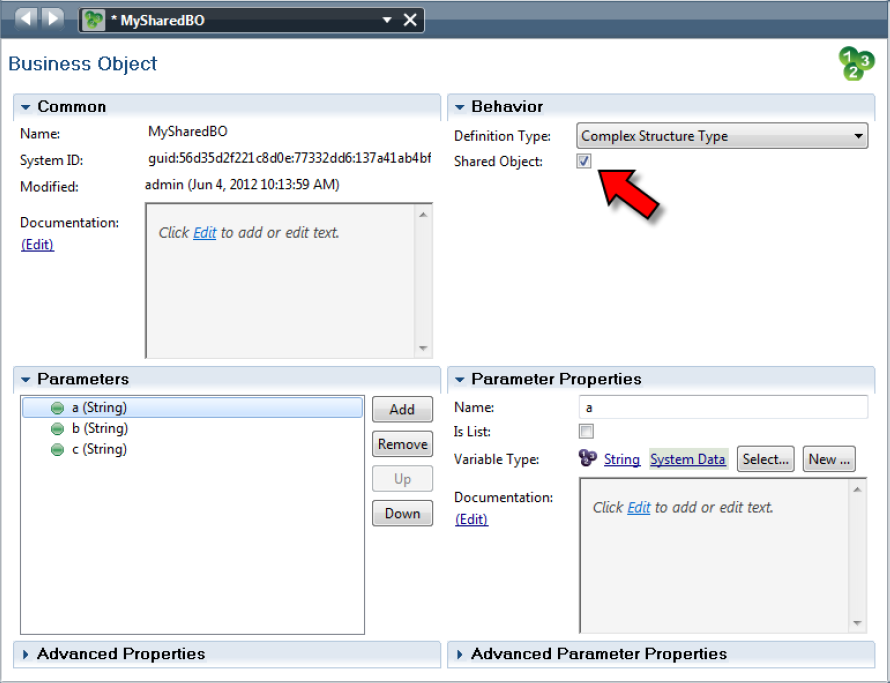
**创建可搜索的变量**

流程入口可以用作找到基于值在变量流程里的流程实例。因为IBPM要做更多的工作去索引出查找的变量值， 这个特征被应用在一个变量通过一个变量基础上。在变量的声明时候， 有一个选项来应用查询业务数据。



**定义和使用共享对象**

从IBPM v8之后， 添加一种共享对象的能力， 共享对象有能力去标记一个业务对象定义为一个共享对象。看接下来的图：

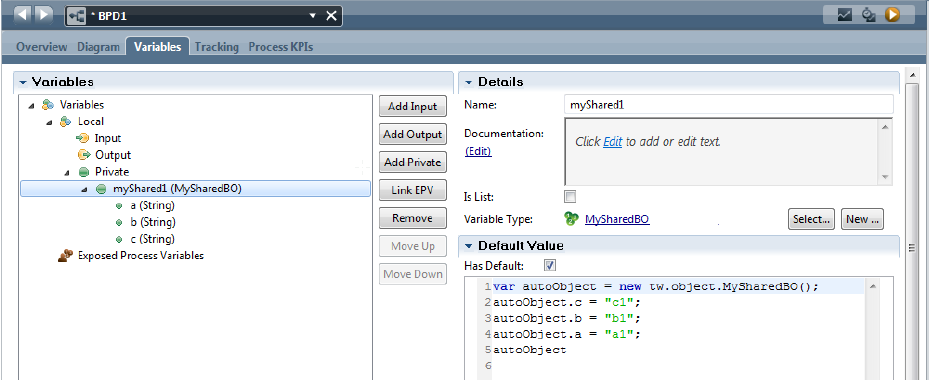


记住标记为共享的业务对象有一个‘绿色’的图标，相对于非共享业务对象的‘紫色’图标。

|  |  |
| --- | --- |
|  | 非共享业务对象 |
|  | 共享业务对象 |

一旦被标记， 现在这种业务类型创造的变量被传递，通过与环境中值传递相反的方式。这就意味着多个流程或者平行的步骤在流程里能同时地‘查看’和‘更新’变量。 如果我们希望有多个流程同时地工作在一个共享对象上，那么我们需要传递一个‘键’，能够用作访问在不同流程中的变量。

这个将会有感觉的，因此让我们尝试并且开始， 考虑之前的截图， 显示了一个新创建的业务对象类型‘MySharedBO’，现在考虑一个流程实例，它定义了一个变量‘MySharedBO’。



我们应该能够简单的假设一个类型为MySharedBO的实例将‘保持在’某个地方作为未来的参考。 假设我们希望有第二个流程利用同一个共享对象。 第二个流程能定义了一个类型为MySharedBO的变量，但…获取这个对象的引用，将要执行：

tw.local.myShared2 = new tw.object.MySharedBO(keyValue);

但是等等…‘ keyValue’参数是从哪来的？

在对业务对象的原始引用上，我们能访问它的‘Key’值用：

var keyValue = tw.local.myShared1.metadata("key");

‘metadata("key")’方法返回一个key的字符串的陈述给那个共享对象。

注意实际上那个‘key’在这次请求是一个显式关键字，并且不是一个变量名。

当一个更改被创建应用在包含在数据共享对象里的数据，那数据被写到一个数据库在场景之后。仅仅当这个数据被写，数据将能够被一个伙伴查看。当一个service的修个完成时或者当一个隐式的请求调用save()方法时，数据被写入。

数据相关的共享对象与一个创建的流程实例相关联， 当一个流程实例数据被清除，共享对象也被清除。

要求进一步的考虑的一个重要区域是什么时候一个流程看见另外一个流程做的数据更改， 答案是不完全，一个可能无能力地期待。假设两个流程实例，P1和P2，P1创建一个共享对象的实例并且设置它的值，然后P2也创建一个引用来自P1 Key的实例。 现在P2更改一个数据的值并且执行一个save（）方法。现在P1能看到什么呢？

选项会显示成：

1. 字段的原始值和原始P1设置的值一致

2. 字段的新值和P2设置的一致

答案是‘看情况’。

这是‘在共享对象最新数据版本’概念。这是什么将会被用到，当任何一个新的有给定的Key的共享对象实例被创建的时候，或者一个存在的实例被一个隐式的load（）方法重载的时候。因此如果P1执行一个load（）方法，它会永远获得最新的数据。

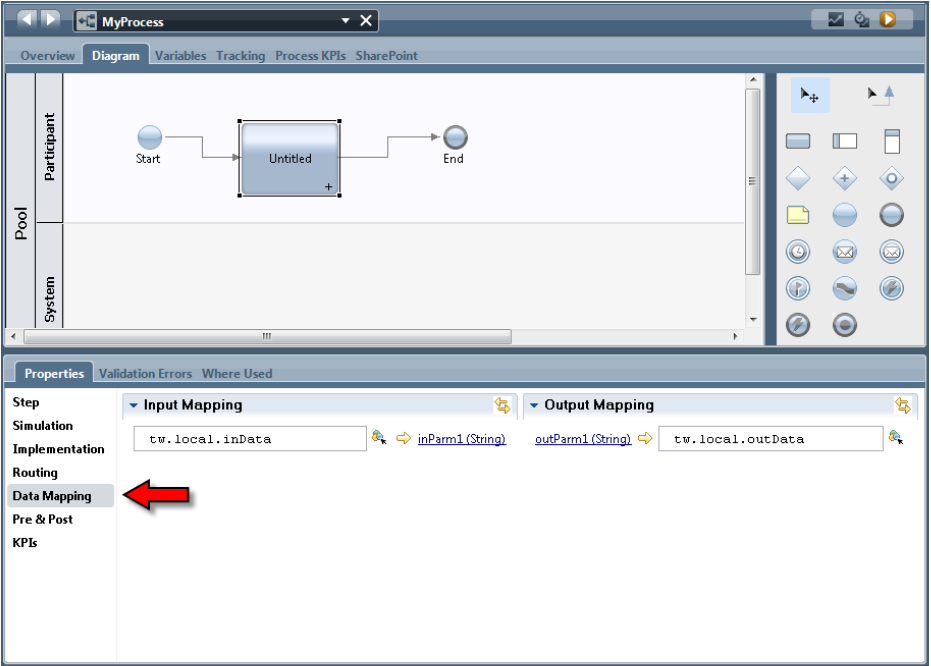
一个P1里重载的数据自动地发生，当流程里的一个步骤被转变或者当一个任务被唤醒。然而，在一个单个直线经过service的上下文里面，变量的值不会改变，除非它被隐式的重载。流程里一步一步的转变是一个显式的重载，因为它是唤醒任务。

**BPD 变量和Service变量-映射**

一个业务流程定义有相关的变量。有些变量可能是流程输入，有些可能是从结束的流程输出和有些可能是只对流程可见的本地变量。

相对一个BPD有的变量，一个IBPM service也可以有。 像BPD一样，IBPM service有输入，输出和本地变量。这些被定义在IBPM Service的定义里。

一个IBPM Service通常被调用在一个BPD经过一个活动节点的上下文里面。当节点被定义，它被一个IBPM Service相关联。如果我们思考这个一会，我们看见这个IBPM Service有期待包括数据的输入变量和会返回某些对于流程一会可用的数据的输出变量。 这些在IBM Service变量需要被‘映射’到变量范围在正在调用Service的BPD里面。



当一个活动在BPD编辑里面被选择， 属性部分显示一个数据映射标签。选择这个显示期待的输入变量对应相关的IBPM Service和生成的输出变量。然后这些能被映射到在当前里的变量。图标在‘盒子’（）旁提供一个定义变量的智能查询，因此我们不需要记住Javascript变量名。

如果一个变量被定义为一个复杂类型， 它将需要在被赋值前被初始化。它的初始化值是‘null’，它没有任何的字段。 一个好的方式去实现初始化是在数据类型定义选上有默认。