

# What is a Business Object?

FlowWright business object (BO) is an object that gives design and real time access to business objects from other systems/application. Think of it as a proxy for business objects. Business objects are implemented using a simple interface "delBusinessObject".



Business objects are configured with a class name, namespace and the DLL file they belong to. Once configured, they become available within the workflow designer for defining objects that are business object type.

After defining these business objects, these business objects can be used within processes to get and set their property values. By using the step "getBusinessObject", the business object can be retrieved at runtime and then its properties used within other computations. Let's say the "Person1" BO has a property called "age". This property can be referenced within other steps such as the "Decision" step to perform a comparison such as the following: *int.Parse("cDevBO.Person1.age")* == 15



### Writing a custom Business Object

A custom BO can be implemented using a the interface "deIBusinessObject". Below is the sample code the clsPersonBO business object.

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.IO;
using FlowWright.cDevDecisionEngine;
namespace FlowWright.cDevBusinessObjects
{
    public class clsPersonBO : deIBusinessObject
    {
        public object getObject(clsEngineContext oEngineContext, Hashtable oInputPar
        ameters)
        {
```

clsPerson oPerson = new clsPerson(oInputParameters["name"].ToString(), int.P
arse(oInputParameters["age"].ToString()));

```
return (oPerson);
        }
        public List<string> getInputParameterKeys()
        {
                List<string> oList = new List<string>();
                oList.Add("name");
                oList.Add("age");
                return (oList);
        }
        public Type getObjectType()
        {
                return (typeof(clsPerson));
        }
        public List<string> getDynamicPropertyList()
        {
                List<string> oList = new List<string>();
                oList.Add("alpha");
                oList.Add("beta");
                return (oList);
        }
public string getDynamicPropertyValue(clsEngineContext oContext, string key)
        {
                return (key + 1.ToString());
        }
```



```
public bool setDynamicPropertyValue(clsEngineContext oContext, string key, s
tring val)
                 {
        File.AppendAllText(@"c:\temp\eventdata\dyna.txt", key + " - " + val);
                         return (true);
                 }
        }
        public class clsPerson
        {
                 public string name { get; set; }
                 public int age { get; set; }
                 public clsData oData { get; set; }
                 public clsPerson(string name, int age)
                 {
                         this.name = name;
                         this.age = age;
                         Random rnd = new Random();
                         this.age = rnd.Next(1, 100);
                         this.oData = new clsData() { num1 = 5, num2 = 7 };
                 }
        }
        public class clsData
        {
                 public int num1 { get; set; }
                 public int num2 { get; set; }
        }
}
```



Once the custom BO is built, place the DLL file within the FlowWright's "bin" folder and use the auto detect feature to auto configure the BO.

| usiness Object | s      | <i>(</i> ) |        |       |                |             |                    |            |
|----------------|--------|------------|--------|-------|----------------|-------------|--------------------|------------|
| ate Import     | Remove | Update     | Export | Table | Auto<br>Detect | Properties  | <b>M</b><br>Inputs | Usage      |
|                |        |            | Edit   |       |                | Viev        |                    | Statistics |
|                |        | 2          |        |       |                | Business Ob | jects              | cieu       |
| 1              |        |            |        |       |                |             |                    |            |
|                | 1      |            |        |       |                |             |                    |            |

## Using the Business Object within a Workflow process

First make sure the above described "clsPeronBO" BO is configured within the Business Objects section of FlowWright's Configuration Manager. Create a new Workflow Definition called "TestBO Def".

| =  | Create Definition   |
|--|---|
| Definitions<br>Definitions<br>Create<br>Ariables | Enter Definition Name:*          TestBO Def         Open Designer |
| Last 10 Updated Last 10 Cro                      | Create  |



Next open the new "TestBO Def" definition within the graphical designer.

| 🐕 cDevWorkflow 🛛 🛛 😽 😽  | =  | Welcome, Admin User   |
|-------------------------|--|-----------------------|
| 🚳 Dashboard             | A Definitions  |                       |
| Lefinitions             | Definitions  |                       |
| 1:1 Instances           | Create Master Variables Participante Remove Remove Copy Snapshots Update Waking Create Copy Snapshots Update Create Waking Create Report Archive Instances | istory                |
| ⊘ Tasks 🝷               | Opens the designer for the     Last 10 Updated Last selected definition     orites All   |                       |
| Forms •                 |  | Search:               |
| 👸 Business Intelligence | Definition Name 💠 Created On 💠 Created By 💠 Updated On   | 🚽 Used By Instances 🗳 |
| 🖌 Steps 👻               | TestB0 Def 12/11/2015 11:39:26 AM Admin User 12/11/2015 11:39:26 AM  | 0                     |

Using the designer toolbar button "Manage Business Objects", let's define a business object to be used within the process.

| Show Labels     All Steps | Home UI                        |                           | T                          | estBO Def - v8.7                         |               |                       |
|---------------------------|--------------------------------|---------------------------|----------------------------|--|---------------|-----------------------|
| Q Search                  | Show Grid Defi<br>Index Varial | e Manage Business Objects | e Import<br>Step Variables | Export Save Snap<br>PNG Definition Manag | shot<br>ement | Delete Comment Search |
|                           |                                |                           | esign                      |  |               | LOIT                  |
| 🤏 🛼                       |                                |                           |                            |  |               |                       |
| SOL                       | start                          |                           |                            |  |               |                       |

Manage Business Objects UI will render within the right pane as follows:

| Manage Business Objects |   |
|-------------------------|---|
| Add BO                  |   |
|                         |   |
| clsPersonBO             | • |
| Add Remove              |   |
|                         | * |
|                         |   |
|                         |   |
|                         | • |
| Used By Steps Save      |   |
|                         |   |



Let's create a process level Business Object called "Person1" that is type "clsPersonBO".

| Manage Business Objects |   | Manage Business Objects |
|-------------------------|---|-------------------------|
| Add BO                  |   | Add BO                  |
| Person1                 |   |                         |
| Туре:                   |   | Туре:                   |
| clsPersonBO             |   | clsPersonBO v           |
| Add Remove              |   | Add Remove              |
|                         |   | Person1:clsPersonBO     |
|                         |   |                         |
|                         | • | -                       |
| Used By Steps Save      |   | Used By Steps Save      |

Now that the Business Object is defined, let's start using the BO within the process. First, is to get the BO using the step "getBusinessObject".





Click the "getbusinessobject" step to configure its properties:

| Step Properties                                    |                             |                                   |   |   |   |
|--|-----------------------------|-----------------------------------|---|---|---|
| Incoming Connections: 1<br>Outgoing Connections: 0 |                             |                                   |   |   |   |
| get Person1 BO                                     |                             |                                   |   |   |   |
| Description  | Cobra/cdevworkflow/deBoProp | s.aspx?fieldName=boProps&cols=Par | ram Name,Param Value - Google Chrome —  |   | × |
|  | cobra/cdevworkflow/de       | BoProps.aspx?fieldName=b          | poProps&cols=Param%20Name,Param%20Value |   |   |
| parms to the business object*                      | Select Business Object:     | Person1:clsPersonBO v             |   |   |   |
|  |                             | Name                              | Value                                   |   |   |
| time to live amount                                | name                        |                                   | Joe I                                   |   |   |
| there be live to a                                 | age                         |                                   | 45                                      |   |   |
| The to live type                                   |                             | Save                              | Cancel                                  |   |   |
| load properties on demand                          | Variables                   |                                   |   | _ |   |
| Log Message  |                             |                                   |   |   |   |
| Documentation                                      |                             |                                   |   |   |   |
| Save   |                             |                                   |   |   |   |

You can also configure few other properties such as whether to load properties on demand and the time to live (ttl) for business object. By configuring time to live or ttl, business object will automatically update itself after the expiration period. So, if you have a BO that has an expiration of 10 minutes, whenever the processes accesses that business object, it will check when the last time the BO was updated, if it has expired, it will refresh itself automatically to provide the latest information to the process.

Since this BO has a property called "age" let's use that property to make a decision. Let's drag a decision step on to the designer canvas and connect to the "getbusinessobject" step.





Let's use the decision step to figure out if the Person1.age is great than 18 years. Click the "decision" step and configure the properties.

| Step Properties                                    |
|--|
| Incoming Connections: 1<br>Outgoing Connections: 0 |
| Name   |
| check age is > 18                                  |
| Description  |
| condition to evaluate*                             |
| int.Parse(cDevBO.Person1.age) > 18                 |
| Log Message  |
| Documentation                                      |
| Save   |

Let's save the changes to the definition and get ready to execute an instance based on this definition.





Navigate to the Workflow Instances menu item and create a new Instance.

| Create Instance                                   |      |
|---|------|
| Select Definition:<br>TestBO Def                  | ▼    |
| By Last Modified By Name By Last Modified By Name |      |
| Enter Instance name:                              |      |
| TestBO Inst                                       |      |
| Execute:  |      |
| None Execute With params                          |      |
|   |      |
| Create  | lose |

Once the Instance is created, select the Instance and execute the Instance.

| 🛃 Instance       | es              |                     |                         |               |             |       |                   |         |                   |       |                   |          |        |                        |                  |         |                |      |
|------------------|-----------------|---------------------|-------------------------|---------------|-------------|-------|-------------------|---------|-------------------|-------|-------------------|----------|--------|------------------------|------------------|---------|----------------|------|
| Instances        | Util            | s                   |                         |               |             |       |                   |         |                   |       |                   |          |        |                        |                  |         |                |      |
| Create Ru<br>Eng | un Re<br>gine F | 9<br>efresh<br>Page | <mark>)</mark><br>Reset | Reset To Step | Remove      | Abort | Update<br>Waiting | Execute | Execute<br>Params | Get   | Execution<br>View | n Render | Report | : View<br>Subworkflows | View<br>Instance | Archive | 🔀 Add 🐋 Remove |      |
|                  |                 |                     |                         | Reset         |             |       | Instances         | Đ       | ecute             | XML   |                   |          | Vie    | W                      |                  |         | Favorites      |      |
| Last 10 Upd      | dated           | My Ins              | stances                 | Favorites     | By Status 🗸 | •     |                   |         |                   |       |                   |          |        |                        |                  |         |                |      |
|                  |                 |                     |                         |               |             |       |                   |         |                   |       |                   |          |        |                        |                  | Search: |                |      |
| In               | stance          | Name                | \$                      | Create        | d On 🌲      |       | Started O         | n 🛊     | End               | ed On | \$                | Status   | \$     | Wait Step              | s 🜲              |         | Definition Nar | ne 🔶 |
| T 100 1          |                 |                     |                         | 12/11/2015 1  | 2.16.16.014 |       |                   |         |                   |       |                   |          |        |                        |                  | T 100 D |                |      |

After executing the Instance, the Instance should have a status of "finish". Click the "Render" button on the toolbar to render the executed Instance.





The rendered Instance will look as follows:

| - Instan          | ices / Rer        | nder Ins             | tance            |                 |       |       |                        |                  |         |                |      |           |  |
|-------------------|-------------------|----------------------|------------------|-----------------|-------|-------|------------------------|------------------|---------|----------------|------|-----------|--|
| Render I          | nstance           |                      |                  |                 |       |       |                        |                  |         |                |      |           |  |
| Execution<br>Only | Execution<br>View | <b>T</b> ask<br>View | View<br>Instance | Tasks<br>Render | Forms | Users | Log<br>Log<br>Messages | Business<br>Data | History | Get<br>XML     | Step | Execution |  |
| TestBO            | Inst              |                      |                  |                 |       |       |                        |                  |         |                |      |           |  |
|                   |                   |                      |                  |                 |       |       |                        |                  |         |                |      |           |  |
|                   |                   |                      |                  |                 |       |       |                        |                  |         |                |      |           |  |
|                   |                   |                      |                  |                 |       |       |                        |                  |         |                |      |           |  |
|                   |                   |                      |                  |                 | •     |       | ]                      |                  |         | $\overline{2}$ |      |           |  |

Click the "check age is > 18" step to drill down into the step information.





#### The decision step should render as follows:

| Step Pro         | perties   |           |          |            |       |        |  |
|------------------|-----------|-----------|----------|------------|-------|--------|--|
| Step Proper      | rties     |           |          |            |       |        |  |
| 1 ▼<br>Execution | Execution | Variables | Business | Properties | Tasks | Errors |  |
| Iteration        | Info      |           | Object   |            |       |        |  |

## **TestBO Inst: Execution Information**

| Step Name         | check age is $> 18$    |
|-------------------|------------------------|
| Step Туре         | decision               |
| Step Status       | complete               |
| Step Return Value | True                   |
| Step Started At   | 12/11/2015 12:21:18 PM |
| Step Ended At     | 12/11/2015 12:21:19 PM |
| Reset To Step     |                        |
|                   |                        |

In reference to the above graphic, the decision step evaluated the expression

'int.Parse("cDevBO.Person1.age") > 18' and returned a value of "True" for the return value of the step. Click the "Properties" button to view the properties for the step.

| efresh Properties from Design |    | Refresh Non-Executed Steps from Design |  |
|-------------------------------|----|--|--|
| condition                     | è, | int.Parse(cDevBO.Person1.age) > 18     |  |
| documentation                 |    |  |  |
| itemDescription               |    |  |  |
| itemName                      |    | check age is > 18                      |  |
| logMessage                    |    |  |  |



Click the little icon next to the condition to view the expression with BO replacement values:

| estBO Inst: Properties         |             |                              |  |  |  |
|--------------------------------|-------------|------------------------------|--|--|--|
| Refresh Properties from Design | Refresh Noi | n-Executed Steps from Design |  |  |  |
|                                |             |                              |  |  |  |
| condition                      | È,          | int.Parse("42") > 18         |  |  |  |
| documentation                  |             |                              |  |  |  |
| itemDescription                |             |                              |  |  |  |
| itemName                       |             | check age is > 18            |  |  |  |
| logMessage                     |             |                              |  |  |  |