

# Dynamic Access Property Binding

Dynamic access binding is an additional binding technique that can be used together with Java Bean binding as described in the previous section. Dynamic access binding does not require an explicit definition of a set/get method for each property but is able to access the properties by generic data access functions.

Adapter objects, as well as embedded objects that are accessed inside the access path, may optionally support the interface `IDynamicAccess`. In this interface, you declare that there are additional properties to be accessed by generic access methods.

This chapter covers the following topics:

- Interface `IDynamicAccess`
  - Example
- 

## Interface `IDynamicAccess`

The interface definition looks as follows:

```
public interface IDynamicAccess
{
    public String[] findDynamicAccessProperties();
    public Class getClassForProperty(String property);
    public void setPropertyValue(String propertyName, Object value);
    public Object getPropertyValue(String propertyName);
    public void invokeMethod(String methodName);
}
```

It informs which dynamic properties are supported. In addition, you have to specify which class of a property it is. You can use any of the classes that are listed in *Appendix C - Data Types to be Used by Adapter Properties* for simple-value properties - and any class you desire for embedded object properties that you follow inside your access path. If you return a null value as a result of the `getClassForProperty()` method, the runtime returns the value as a String object.

Besides, you have to implement the generic set and get functions for property access.

There is a generic method invoked, e.g. when the user chooses a button in the page bound to a dynamically called method.

## Example

The following example shows an adapter object that has two Bean properties (`firstName`, `lastName`) and three dynamic properties (`street`, `city`, `birthday`):

```
// This class is a generated one.

import com.softwareag.cis.server.Adapter;
import com.softwareag.cis.server.IDynamicAccess;
import com.softwareag.cis.util.CDate;
```

```

public class DynamicAccess_Adapter
    extends Adapter
    implements IDynamicAccess
{
    String m_firstName;
    String m_lastName;
    String m_street;
    String m_city;
    CDate m_birthday;

    public String[] findDynamicAccessProperties()
    {
        return new String[] {"street","city","birthday"};
    }

    public void setPropertyValue(String propertyName, Object value)
    {
        if (propertyName.equals("street")) m_street = (String)value;
        else if (propertyName.equals("city")) m_city = (String)value;
        else if (propertyName.equals("birthday")) m_birthday = (CDate)value;
        else throw new Error("No property " + propertyName + " available");
    }

    public Object getPropertyValue(String propertyName)
    {
        if (propertyName.equals("street")) return m_street;
        if (propertyName.equals("city")) return m_city;
        if (propertyName.equals("birthday")) return m_birthday;
        throw new Error("No property " + propertyName + " available");
    }

    public Class getClassForProperty(String propertyName)
    {
        if (propertyName.equals("birthday")) return CDate.class;
        // default: null ==> String is assumed by runtime
        return null;
    }

    public void invokeMethod(String methodName) {}

    // -----
    // bean properties
    // -----
    public void setFirstName(String value) { m_firstName = value; }
    public String getFirstName() { return m_firstName; }
    public void setLastName(String value) { m_lastName = value; }
    public String getLastName() { return m_lastName; }
}

```

Specifying a layout definition, there is no difference between the dynamic access properties and the bean properties.

The screenshot shows a window titled "Dynamic Access" with a close button in the top right corner. Below the title bar is a "Save" button. The main area is a "rowarea" titled "Address Data" containing five input fields: "First Name", "Last Name", "Street", "City", and "Birthday". The "Birthday" field is a dropdown menu.

The layout definition for the above page looks as follows:

```
<page model="DynamicAccess_Adapter">
  <titlebar name="Dynamic Access">
    </titlebar>
  <header withdistance="false">
    <button name="Save">
    </button>
  </header>
  <pagebody>
    <rowarea name="Address Data">
      <itr>
        <label name="First Name" width="100">
        </label>
        <field valueprop="firstName" length="20">
        </field>
      </itr>
      <itr>
        <label name="Last Name" width="100">
        </label>
        <field valueprop="lastName" length="20">
        </field>
      </itr>
      <itr>
        <label name="Street" width="100">
        </label>
        <field valueprop="street" length="20">
        </field>
      </itr>
      <itr>
        <label name="City" width="100">
        </label>
        <field valueprop="city" length="20">
        </field>
      </itr>
      <itr>

```

```
        <label name="Birthday" width="100">
        </label>
        <field valueprop="birthday" length="20" datatype="date">
        </field>
    </itr>
</rowarea>
</pagebody>
<statusbar withdistance="false">
</statusbar>
</page>
```