Customized Proposals - Example

Customized Proposals - Example

The address maintenance example from the beginning of this part is now shown in detail. Since it is an extension of the previous example for the customized layout, only the additions are shown that are responsible for the management of proposal values.

This chapter covers the following topics:

- XML Layout
- Java Adapter Code
- Directly Accessing Proposal Values

XML Layout

The XML layout definition is:

```
<page model="CustomizedLayoutAdapter">
    <titlebar name="Address Edit">
        <persedit persprop="paInfo">
        </persedit>
    </titlebar>
    <header withdistance="false">
        <button name="Save" method="onSave">
        </button>
    </header>
    <pagebody>
        <rowarea name="Address" visibleprop="adressAreaVisible">
            ...
            <itr visibleprop="townVisible">
                <label name="Town" width="100">
                </label>
                <field valueprop="town" width="200">
                </field>
            </itr>
            <itr visibleprop="countryVisible">
                <label name="Country" width="100">
                </label>
                <field valueprop="country" width="50">
                </field>
            </itr>
            ...
        </rowarea>
    </pagebody>
    <statusbar withdistance="false">
    </statusbar>
    <personalization>
        <persproposal property="town" comment="Town">
        </persproposal>
        <persproposal property="country" comment="Country">
        </persproposal>
        ...
        </persfilter>
    </personalization>
</page>
```

Below the PERSONALIZATION tag, you see two PERSPROPOSAL tags. Each tag holds the following information:

- The name of the property that should be proposed.
- A comment.

Java Adapter Code

The adapter code is:

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

public class CustomizedLayoutAdapter
    extends Adapter
{
    PersonalizationAdapterInfo m_paInfo = new PersonalizationAdapterInfo(this);
    ...

    ...

    public String getTown() { return m_town; }
    public void setTown(String value) { m_town = value; }

    public String getCountry() { return m_country; }
    public void setCountry(String value) { m_country = value; }
...
    public void init()
    {
        // switch maintenance on
        Personalization.switchPersonalizationMaintenanceOn(findSessionContext());
        // set up scenarios
        PersonalizationScenarioSequence pss = new PersonalizationScenarioSequence("customer");
        Personalization.defineScenarioSequenceInContext(findSessionContext(),pss);
        // transfer proposal values
        m_paInfo.applyProposals(this);
    }
...
}
```

You see that inside the `init()` method, the method `applyProposals(...)` is called. This method is responsible for transferring the proposal values into the corresponding properties of the adapter. For the transfer, just the normal methods are used: i.e. properties are either set via their corresponding set method or by calling the `IDynamicAccess` interface.

It is completely up to you where to embed the `applyProposals(...)` into your adapter code. To put it into the `init()` method is just one example. Maybe you want to make the decision whether to propose values or not dependent on some other conditions inside your program: when you create a new address, you want to propose values - however, when you edit an existing address, you do not want to propose values.

Directly Accessing Proposal Values

You can also access proposal values directly. The `PersonalizationAdapterInfo` class offers a corresponding method `getAllProposalValues()` to do so:

```
public void onDirectAccess()
{
    Properties props = m_paInfo.getAllProposalValues();
    if (props == null)
    {
        outputMessage(MT_ERROR,"No proposal values are available");
        return;
    }
    Iterator keys = props.keySet().iterator();
    while (keys.hasNext())
    {
        String key = (String)keys.next();
        String value = props.getProperty(key);
        System.out.println("Key/Value: " + key + "/" + value);
    }
}
```