Technical Note

Loading a Choice List Based on a Parameter

Author(s): R&D Department

Publication date: September 14, 2006

Revision date: May 2010

Loading a Choice List Based on a Parameter

Overview

It is quite common that we need to load a dynamic choice list based on a parameter specified in the WHERE clause of its associated SQL command. When the choice list depends on such a parameter, it is not possible to determine in advance the correct list, and therefore errors can occur.

REP++ greatly facilitates the loading of dynamic choice lists that do not depend on the value of a parameter. These lists are loaded once at the beginning of the program and they contain all the possible values for their associated field.

But when a choice list depends on a parameter:

- It cannot be loaded until the parameter value is known, and
- It must be refreshed every time the parameter value changes.

This article describes how to how to load a choice list based on a parameter when the parameter is:

- A field that is part of a Rowset other than the Rowset that loads the choice list.
- A field that is part of the same Rowset that loads the choice list.
- Not a field (e.g. a session variable).

Parameter is a field in a Rowset other than the Rowset that loads the choice list

In this case, you have a field (**MyField**) in a Rowset (**MyRowset**). **MyField** has a choice list whose content depends on a parameter that is a field (**ParamField**) in another Rowset (**ParamRowset**). All you have to do to filter the choice list based on the value of this parameter is to add the following WHERE clause to the choice list's SQL command:

```
SELECT … WHERE NameOfCol=:ParamRowset.ParamField
```

At run time, REP++ retrieves the value of **ParamField** and binds it automatically to the SQL command of the choice list.

Please note that in order for this approach to work, the content of **ParamRowset** must have been read before the choice list of **MyRowset.MyField** is loaded.

Parameter is a field in the same Rowset that loads the choice list

In this case, you have a field (**MyField**) in a Rowset (**MyRowset**). **MyField** has a choice list whose content depends on a parameter that is found in a field (**ParamField**) within **MyRowset**.

The issue in this case is the following circular dependence:

- You cannot read a line from the database to **MyRowset** because the choice list of **MyField** is not loaded.
- You cannot load the choice list of **MyField** until **ParamField** gets a value.
- **ParamField** cannot get a value until a line is read from the database into **MyRowset**.

To illustrate the problem, let's say that you have an application that contains a rowset EMPLOYEE, with the following fields: ID, DEPARTMENT and SALARY_RANGE. The field SALARY_RANGE has a choice list that depends on the DEPARTMENT. In order to read an employee's data, the correct list of salary ranges must be known; this requires knowing the employee's department, which means reading the employee's data. At the same time, the employee's data cannot be read until the list of salary ranges is loaded.

To break this circular dependence, you need to be able to read from the database without imposing a restriction on the value to be read into **MyField**.

This method requires that the type of **MyField** in the repository must NOT be *Num Simple Choice, Alpha Simple Choice, Num Multi Choice* or *Alpha Multi Choice*. This step allows you to put values in the field that are not part of its choice list.

To load the choice list:

1. Create an SQL atom to retrieve the content of the choice list. The SQL command will be almost identical to the one in the previous section:

```
SELECT … WHERE NameOfCol=:MyRowset.ParamField
```

2. When you add **MyField** to **MyRowset**, select the name of the SQL command that you created in the previous step in the list **SQL Command for List**. Do not check the **ChoiceList by SQL Cmd (...)** option. This will allow you to delay the reading of the choice list until **ParamField** is read.
3. Since you instructed REP++ not to load the choice list automatically, you will have to load it programmatically once **ParamField** is read. Here is a sample code that uses the REP++*framework* for ASP.NET:

```
public override void OnAfterLoadFromDB(CancelEventArgs e) {
    // Put user code to handle this event here
    base.OnAfterLoadFromDB (e);
    // fldMYFIELD is a RepPP.Toolkit.Web.FldDropDownList
    // that is associated with MyField field
    fldMYFIELD.Field.ReadChoiceList();
    // Instruct the control to update its list
    fldMYFIELD.Refresh();
}
```

Parameter is not a field

In this case, you load your choice list programmatically using an SQL command. Here is a code sample that uses the REP++*framework* for ASP.NET:

```
public override bool OnBeforeLoadFromDB(CancelEventArgs e) {
  string strSqlCmd;
  strSqlCmd = "SELECT NAME FROM CLIENT WHERE CODE = 'LULUCY'";
  fldMYFIELD.Field.ChoiceList.BuildFromDb(strSqlCmd);
  fldMYFIELD.Refresh();
  return base.OnBeforeLoadFromDB (e);
}
```

Technical Note

The SQL command was hard-coded in the example above. You should maintain all your SQL commands in the REP++*repository*.