

DBCOMBO

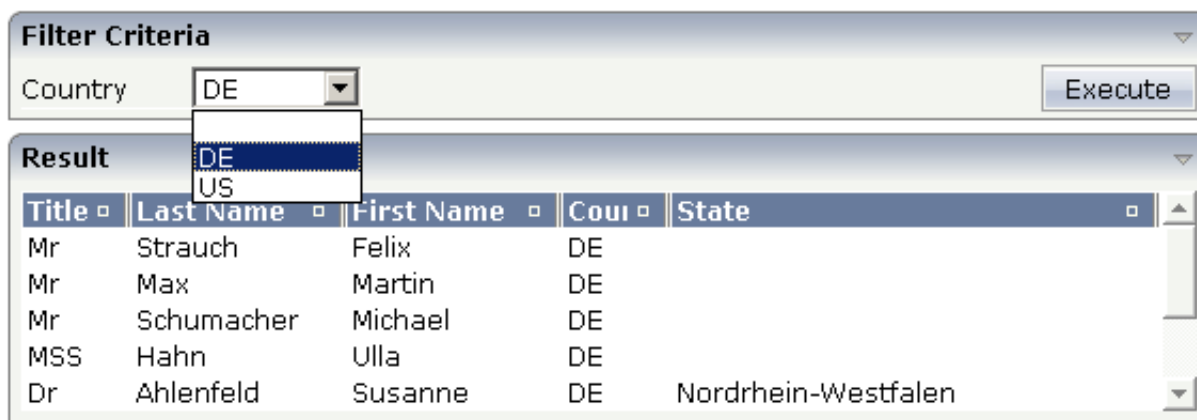
The DBCOMBO control represents a filter criterion of a database query. It provides for a value help that is read from the database and a convenient way to append the filter criterion to the SELECT statement. In contrast to DBFIELD, the valid values are read from the database when the page is loaded.

This chapter covers the following topics:

- Example
- Properties

Example

The following image shows an example in which a DBCOMBO control is used for the filter criterion "Country" within a simple business partner report. The combo box options are read from table "COUNTRY" when the page is loaded.



Have a look at the XML layout definition:

```

<rowarea name="Filter Criteria">
  <itr>
    <label name="Country" width="80">
      </label>
      <dbcombo valueprop="dbComboCountry" querycolumn="COUNTRY" datasource="addressdb"
        valuehelptable="COUNTRY" valuehelpcolumn="ID" width="80">
      </dbcombo>
      <hdist width="100%">
      </hdist>
      <button name="Execute" method="onExecute">
      </button>
    </itr>
  </rowarea>
</vdist>
</vdist>
<rowarea name="Result" height="140">
  <itr height="100%">
    <textgridsss2 griddataprop="result" rowcount="5" width="100%">
      <column name="Title" property="TITLE" width="50">
      </column>

```

```

        <column name="Last Name" property="LASTNAME" width="100">
        </column>
        <column name="First Name" property="FIRSTNAME" width="100">
        </column>
        <column name="Country" property="COUNTRY" width="50">
        </column>
        <column name="State" property="STATE" width="100%">
        </column>
    </textgridsss2>
</itr>
</rowarea>

```

The corresponding adapter code is:

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;

import com.softwareag.cis.server.Adapter;
import com.softwareag.cis.server.util.DBCOMBOInfo;
import com.softwareag.cis.server.util.DBTEXTGRIDCollection;
import com.softwareag.cis.server.util.DBUtil;
import com.softwareag.cis.server.util.DelegateError;
import com.softwareag.cis.server.util.IDBCondition;
import com.softwareag.cis.server.util.IDBQUERYConnectionProvider;

// This class is a generated one.

public class DBCOMBO_Adapter
    extends Adapter
    implements IDBQUERYConnectionProvider, IDBDemoAdapter
{
    // -----
    // members
    // -----

    private int m_port = 9001;
    private int m_dbPort = 9001;
    private Connection m_connection;
    ...
    DBQUERYStartDatabaseThread m_startDemoDBThread;
    // -----
    // property access
    // -----

    // property >port<
    public void setPort(int value) { m_port = value;}
    public int getPort(){ return m_port;}

    // property >dbComboCountry<
    DBCOMBOInfo m_dbComboCountry = new DBCOMBOInfo(this);
    public DBCOMBOInfo getDBComboCountry() { return m_dbComboCountry; }

    // property >result<
    DBTEXTGRIDCollection m_result = new DBTEXTGRIDCollection();
    public DBTEXTGRIDCollection getResult() { return m_result; }
    public void setResult(DBTEXTGRIDCollection value) { m_result = value; }
    ...
    // -----
    // public adapter methods
    // -----

    public Connection getDBConnection(String datasource)
    {

```

```

try
{
    if (m_connection == null)
    {
        String jdbcDriverClassName = "org.hsqldb.jdbcDriver";
        String jdbcUrl = "jdbc:hsqldb:hsqldb://localhost:"+m_dbPort;
        String user = "sa";
        String password = "";
        Class.forName(jdbcDriverClassName);
        m_connection = DriverManager.getConnection(jdbcUrl, user, password);
    }
    return m_connection;
}
catch (Exception exc)
{
    throw new DelegateError(exc);
}
}
...
public void onExecute()
{
    try
    {
        StringBuffer sb = new StringBuffer();
        sb.append("SELECT * FROM BUSINESSPARTNER INNER JOIN ADDRESS ON BUSINESSPARTNER.ID =
                ADDRESS.BUSINESSPARTNERID");
        DBUtil.addToQuery(sb, new IDBCondition[] { m_dBComboCountry }, true);
        String dataSource = m_dBComboCountry.getDataSource();
        Connection con = getDBConnection(dataSource);
        ResultSet rs = con.createStatement().executeQuery(sb.toString());
        m_result.initWithResultSet(rs);
    }
    catch (Exception exc)
    {
        throw new DelegateError(exc);
    }
}
}
}

```

The property `dBComboCountry` is of type `DBCComboInfo` (from the package `com.softwareag.cis.server.util`). The `DBCComboInfo` implements (like all DB controls) the interface `IDBCondition` (`com.softwareag.cis.server.util`). With class `DBUtil` (`com.softwareag.cis.server.util`), you can append values of the filter criteria to the `SELECT` statement in a convenient way. See the JavaDoc documentation of class `DBUtil` for details.

The `DBCComboInfo` class does not open a database connection on its own (same to all DB controls). The embedding adapter provides for an implementation of interface `IDBQUERYConnectionProvider` (`com.softwareag.cis.server.util`) when creating a `DBCComboInfo` object. The interface method `getDBConnection` is called once - at the first time the `DBCComboInfo` has to access the database. There are no updates (insert/update/delete) done with this connection. As the `DBCComboInfo` does not open the connection, it does not care about closing the connection.

For displaying the result, the class `DBTEXTGRIDCollection` (from the package `com.softwareag.cis.server.util`) is used. This class extends `TEXTGRIDCollection` by the ability to initialise the collection with a result set (`java.sql.ResultSet`). For each line of the result set, it creates an object of class `DBTEXTGRIDLine` (package `com.softwareag.cis.server.util`). Class `DBTEXTGRIDLine` implements the interface `IDynamicAccess`. With this, "normal" text grid controls can be used to visualize the data of the `DBTEXTGRIDCollection`.

Properties

Basic			
valueprop	Property that returns a DBCOMBOInfo -instance. This instance provides for the combo box options that are read from database as well as for a convenient way to append the filter value to query string.	Obligatory	
querycolumn	Name of the column in the query to that the filter criteria is belongs to. This column may differ from the value help table/column (properties VALUEHELPTABLE, VALUEHELPCOLUMN). This name is used to build a SQL string in method "toSQLString".	Obligatory	
datasource	Logical identifier of the data source to use. This name is passed to the connection provider in method "IDBQUERYConnectionProvider.getDBConnection".	Obligatory	
valuehelptable	Name of the table from there the list of valid values can be read.	Obligatory	
valuehelpcolumn	Name of the column from there the list of valid values can be read.	Obligatory	
valuehelpcolumndescr	Name of a column where an additional description of the valid values is stored. The name must identify a column inside the "value help table" (property VALUEHELPTABLE).	Optional	
size	Integer value that defines the number of lines being displayed. If the SIZE is set to "1", the selection is displayed as combo box. If it is set to ">1", it is displayed as multi line selection.	Optional	
flush	Flushing behaviour; please view "Common Rules" for details	Optional	screen server
displayonly	If set to "true", the DBCOMBO will not be accessible for input. It is just used as an output field.	Optional	

width	Width of DBCOMBO in pixels or as percentage value.	Optional	100 120 140 160 180 200 50% 100%
align	Explicit Alignment	Optional	left center right
valign	Explicit Alignment	Optional	top middle bottom
colspan	Number of columns occupied by this control.	Optional	1 2 3 4 5 50 int-value
rowspan	Number of rows occupied by this control.	Optional	1 2 3 4 5 50 int-value

<p>combostyle</p>	<p>Explicit style information passed to the DBCOMBO. Example: if you want the text inside the DBCOMBO to be right aligned, define "text-align: right".</p>	<p>Optional</p>	
<p>helpid</p>	<p>Identifier that is used for building the URL of the online help page. Please refer to "Online Help Management" for details.</p>	<p>Optional</p>	
<p>comment</p>	<p>Comment without any effect on rendering and behaviour. The comment is shown in the layout editor's tree view.</p>	<p>Optional</p>	