List Box

Overview

The list box acts like an expanded Select Box. The list box can be used for a single choice, or multiple choices from a list. The list of choices can be loaded in a variety of ways, please see below for more detail.

Field Binding Dialog (Rich UI Only)

Please visit the Field Binding page.

List Box Properties

select box height - This property is used to control the size of the list box and set the number of choices displayed in a selectable list instead of the dropdown box.

multiple - This specifies that multiple values can be selected from the list box. The default value of the property is 'false'. When set to 'true' the selected options are returned to the program as a comma separated list.

Choices / Choice Values Properties

These properties can be used to set the list of options in the list box:

- ***choices property*** - Is where your set the options in order how you would like the list box to display them.
- ***choice values property*** - Is where the value you want returned to your application is set.

One option will appear in the box for each entry in the 'choices' list. The value for each choice will be taken from the corresponding entry in the 'choice values' list. If 'choice values' is not specified, the 'choice' text will also be used as the value.

These fields can also be bound to an RPG field (**Rich UI only**).

There are two ways to populate the choice option/values properties:

1. ***Comma separated list*** - A simple list of values separated by just a comma (i.e. Option 1, Option 2, Option 3, etc...).
2. ***JSON array format*** - Using JSON array format, meaning the choices/values are enclosed in a bracket with quotations and then comma separated (i.e. ["ElementOne", "ElementTwo", "ElementThree"]) within the choices and choice values fields also allows you to populate the auto-complete.

Database-Driven Selection

This section allows choices/choice values to be retrieved from a database file.

Database-Driven Selection	
choices database file	CATEGP
choice options field	CNAME
choice values field	CATID
choices selection criteria	
choices parameter value	
blank option	
blank option label	
order by	
max choices	

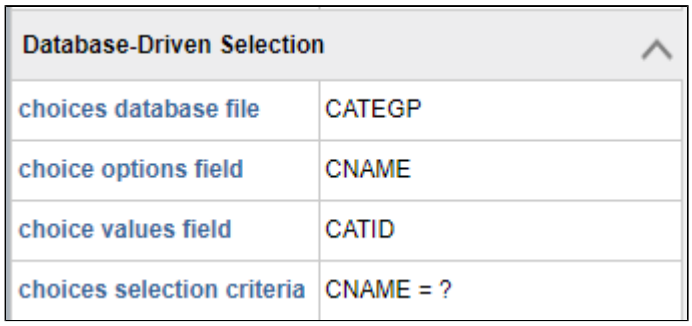
Choices database file - Enter the database file to use to populate your options/values. When setting your database file it's not required to qualify with library name, although you can. if the library is omitted, the application job's library list will be used.

Choice options field - Select the field you wish to use to populate the values shown as choice options.

Choice values field - Select the field to return a value to your application.

In the example, records will be loaded from file CATEGP. The options will be populated by field CNAME. The value in field CATID in the selected record will be returned to the program.

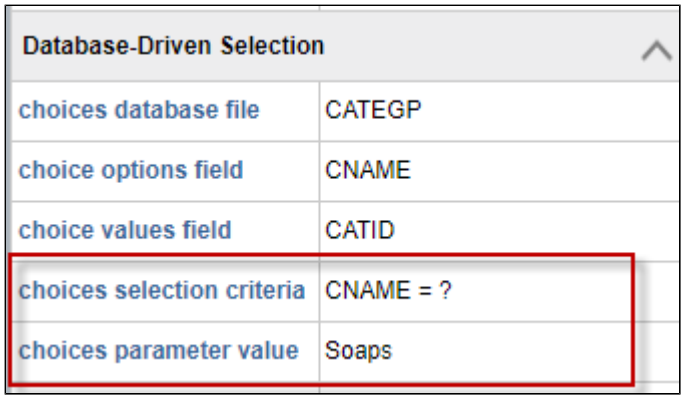
Choice selection criteria - This acts as a WHERE clause in an SQL statement to filter your criteria. SQL parameter markers are always specified by a question mark (?). The property "choices parameter value" can be used to provide the value for this parameter marker.



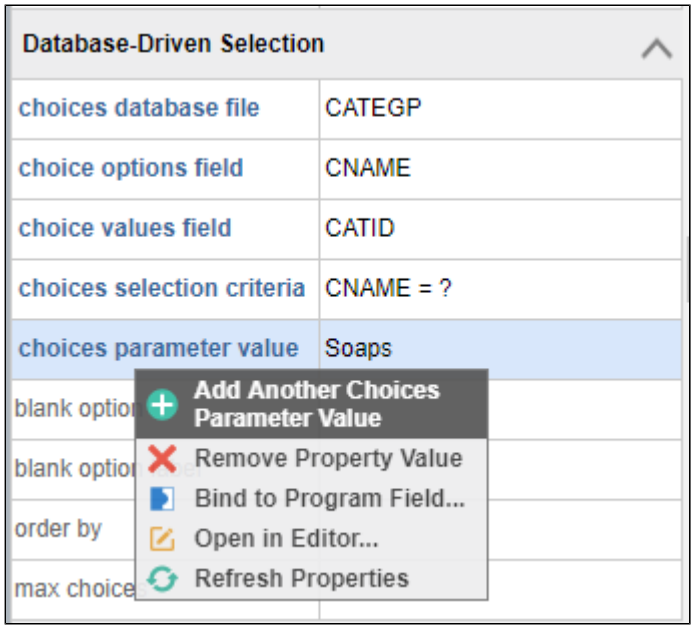
This will produce an SQL statement like this:

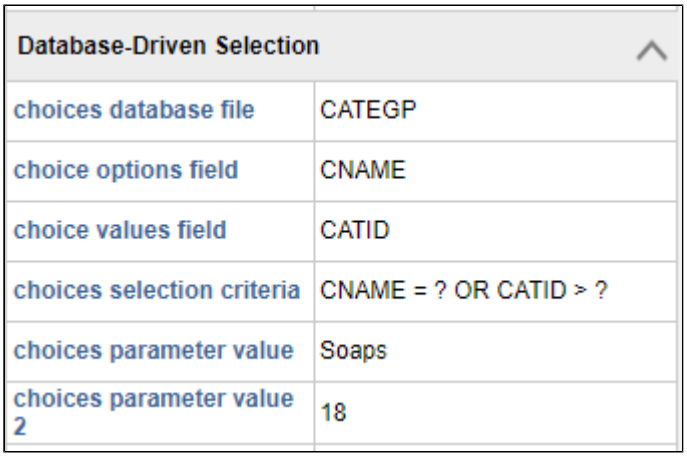
```
SELECT DISTINCT CNAME, CATID FROM CATEGP WHERE CNAME = ?
```

Choice parameter value - This is where the choice selection criteria value for '?' is set. This value may also be bound to a field **(Rich UI only)**.



By completing the choice parameter value we can see our statement should return only the items named "Soaps". You can have multiple "parameter value" properties, however this does require multiple choices selection criteria. These can be added by right-clicking the choices parameter value property name and selecting "Add Another Choices Parameter Value":





For Rich Display Files the choices parameter values would usually be bound to a field allowing dynamic SQL selection.

To find more information on the choice selection criteria and choices parameter properties, view the Parameter Markers section here.

Blank Option

The blank option property allows you to provide a option containing blank text before any other choices from the database file are displayed.

Blank Option Label

By default the blank option label contains no text. This property allows a specific text to be provided to display in the initial blank option. If the initial 'blank' option is submitted with the alternate text the value will be sent to the program as a blank value.

Both ***'blank option'*** and ***'blank option label'*** properties can only be used with a Database-Driven selection. These properties may also be assigned to a bound field. **(Rich UI only)**.

Max Choices
The max choices property allows you to set the number of choices you will see from your database-driven selection. There is no limit to the number of choices you can show, however if there is no limit set the default is 10.

Dynamic Auto-Complete

Choices URL

The choices URL property allows you to use an external program to return your choice options and values by passing them using JSON formatting. PHP scripts or Universal Display Files are examples of custom programs used to pass the values to your application. When using the choices URL property, all database-driven auto-complete properties are ignored.

JSON Successful Response Example

These are values returned from an auto-complete list box in a successful response from the external program:

Format for PUI Version 6, Fix Packs after 1.2

```
{
     "success":true,
     "response":[
          {"text":"Dishes and Cups", "value":"18"},
          {"text":"Soaps", "value":"14"}
     ]
}
```

Legacy Format

```
{
      "success":true,
      "response":[
           new Option("Dishes and Cups","18"),
           new Option("Soaps","14")
      ]
}
```

In the successful response you can see the choices are being set by the values returned. Also in the example above the first value returned is the choice text, and the second returned value is the numeric value associated with the corresponding choice.

JSON Error Response Example

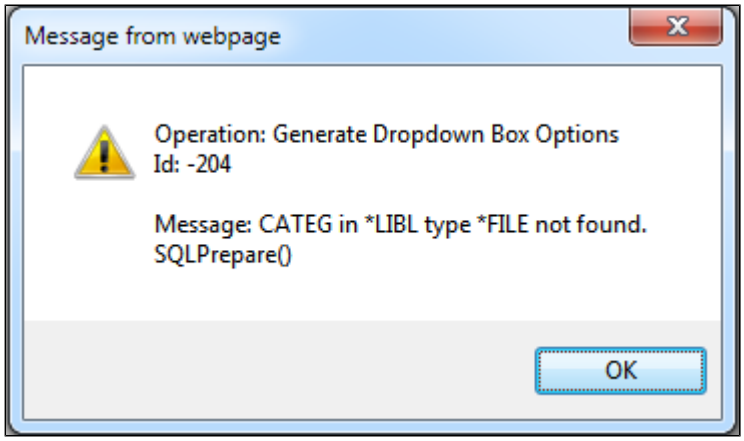
This example is JSON information returned when the script encounters an error with an database driven auto-complete.

```
{
      success:false,
      "errorId":"-204",
      "errorText":"CATEG in *LIBL type *FILE not found.",
      "errorText2":"SQLPrepare()"
}
```

If you are unable to get any information to display on your choice URL script you can use the showErrors() API to try and debug the reason the results are unable to display.

showErrors() API Documentation

Example of the showErrors() API displaying JSON error response example:


Message from webpage
Operation: Generate Dropdown Box Options
Id: -204
Message: CATEG in *LIBL type *FILE not found.
SQLPrepare()
OK