

Block Unwanted Web Pages – Use White List

The web browser performer shipped with Limnor releases provides many ways for you to block unwanted web pages. This article introduces one of such methods. Please read other articles for other methods.

In this sample application, we use a “white list” to allow only showing web pages in the list, and block all other web pages.

[How to block pages](#)

[Block Conditions – Use a White List](#)

[Create a database](#)

[Add Search Table](#)

[Create Database Search Actions](#)

[Set navigation condition](#)

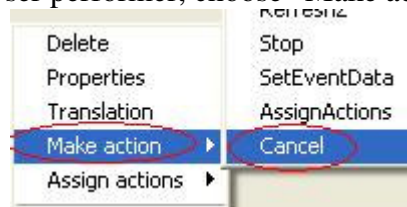
[Put all actions together](#)

[Assign Action List to Events](#)

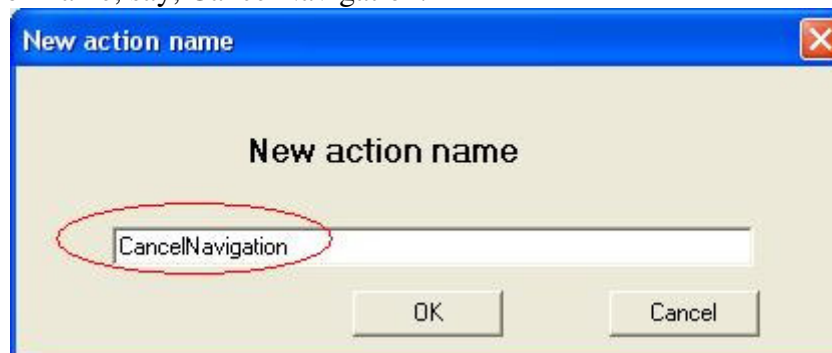
How to block pages

Suppose we already created a web browser performer on a page. A web browser performer has a “Cancel” method. We may use it to make an action to block page navigation.

Right-click on the web browser performer, choose “Make action”, choose “Cancel”:



Give an action name, say, CancelNavigation:



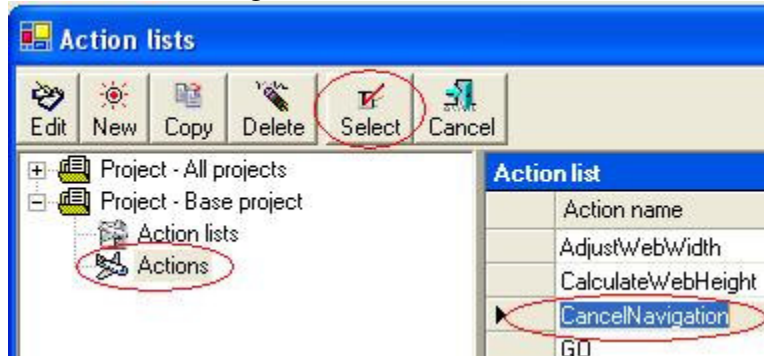
Now this CancelNavigation will block page navigation. The next question is the timing: when is this action carried out? The answer is that this action should be carried out at the events BeforeNavigate and NewWindow. The event BeforeNavigate occurs before the web browser going to a new page. The event NewWindow occurs before the web browser pops up a new window showing a web page.

Suppose we want to block the web browser from going to a new page. Right-click on the web browser performer, choose “Assign actions”, choose “BeforeNavigate”:

Web Browsing Limit to URL List



Select Actions, select CancelNavigation, click “Select” button:



The above programming assigns action CancelNavigation to event BeforeNavigate. This way, we block page navigation on the web browser.

To block pop-up window, we just assign action CancelNavigation to event NewWindow. Right-click on the web browser performer, choose “Assign actions”, Choose “NewWindow”:



Just as we did before, we select action CancelNavigation for it. Now all pop-up windows will be blocked.

Block Conditions – Use a White List

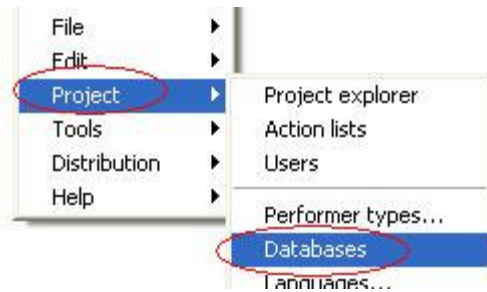
If we run the application now, we will see that the web browser will not show any web pages. This is because we blocked all pages. This is certainly not we wanted. We may just want to block certain pages, not just allow certain pages, or use other criteria to determine which pages to block.

In this sample, we want to block all pages that are not listed in a database table. We may add all permitted web domains to this table. We use a Microsoft Access database for this purpose.

Create a database

Choose “Project | Databases” menu to manage database connections used for the current application:

Web Browsing Limit to URL List



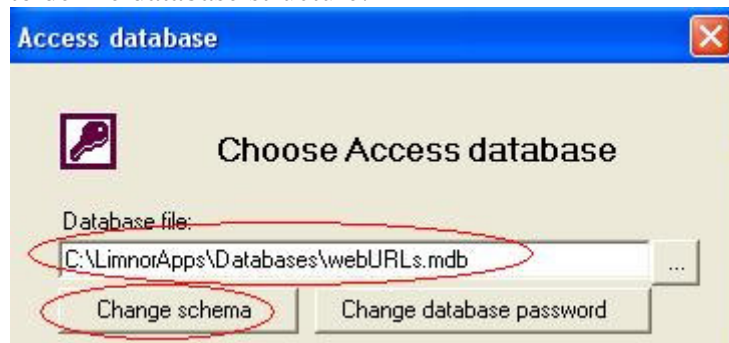
You will see all the database connections listed. Right now, this list is empty. Click “Add” button to add a new database connection:



Give a connection name for this new database connection. You may use any name. Click “Select an Access database” button to specify the Access database file:

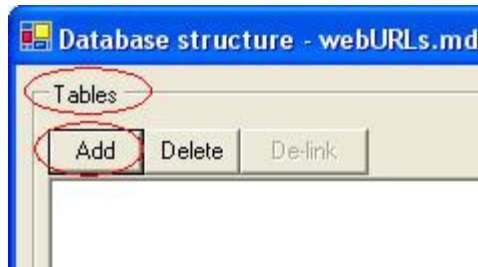


You may type in the database file name, or click ... button to find the file name. The file does not need to exist. If the file does not exist, Limnor will create it. Click “Change schema” button to define database structure:



Click “Add” button under “Tables” to add a new table to the database:

Web Browsing Limit to URL List



Give a table name. Here we call this table "Unwanted":



We want to add a field to this table to hold the web domain we want to block. Click "Add" button under Fields:

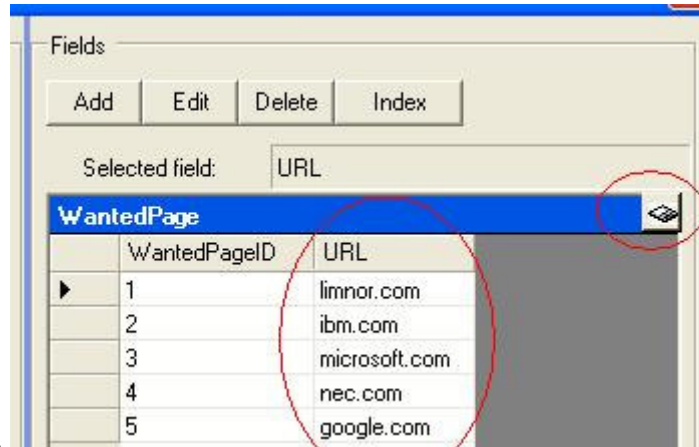



We name this new field as "URL". Field type is "String". That means this field is a serial of characters. We use 200 as Field size. That means the maximum number of the characters for this field is 200:



Now we have the table to record all permitted web domains. We may enter them in the URL field one by one:

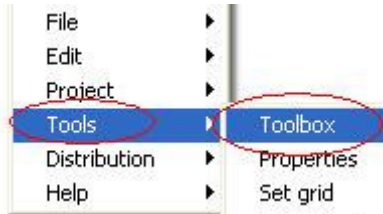
Web Browsing Limit to URL List



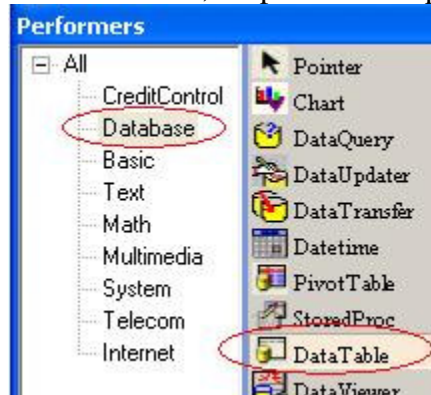
After you enter all permitted URLs, do not forget to click the save button . If you do not click the save button, when the window closes, all data you entered will be lost. In this sample, when the web browser is going to show a new web page, we search this table to see if the web address for the new web page contains a string in the URL field of this table. If it does, we allow the new web page to show up. For example, we will allow web pages like <http://www.limnor.com/Index.html>, <http://www.ibm.com/us>, <http://www.nec.com/>, etc., to appear. In the next section, we show how to use Data Table performer to do the search.

Add Search Table

Right-click on the page, choose “Tools | Toolbox” menu:

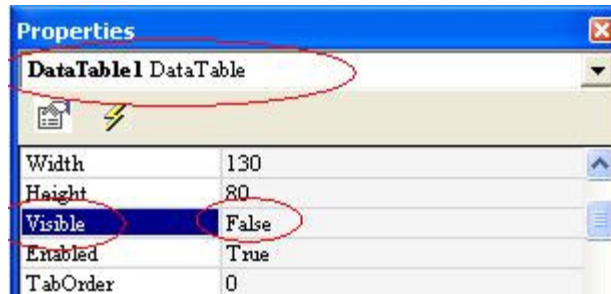


Tool box window appears. Select Database, drop DataTable performer to the page:



Because we do not want to display this table on the page, we need to set its “Visible” property to false:

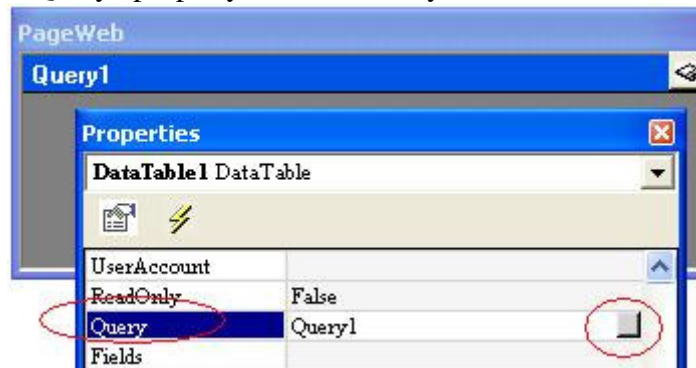
Web Browsing Limit to URL List



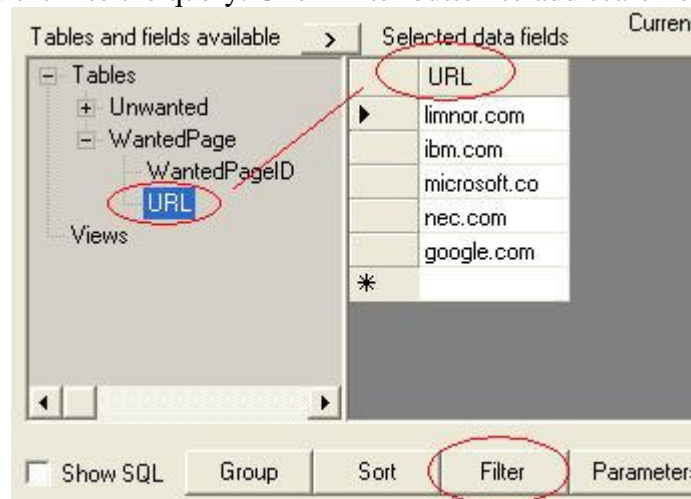
If the Visible property is set to False, at runtime, this performer is invisible. At design time, a grid will appear in a separate window for you to see the data:



We need to set its "Query" property so that we may search the data:

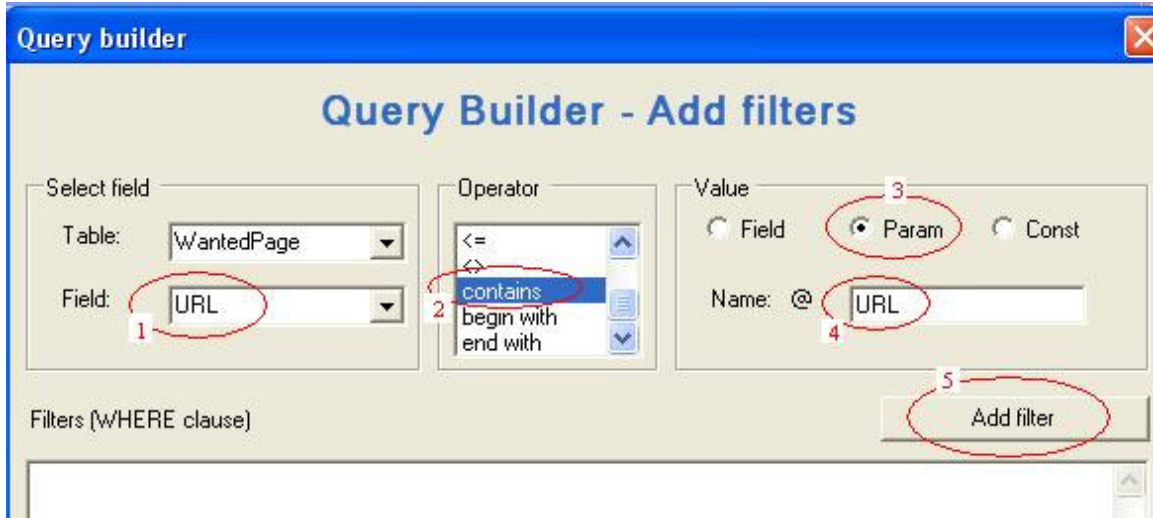


When setting Query property, Query Builder appears. Double-click the UnwantedID and URL fields to add them to the query. Click Filter button to add search condition:



Select the URL field because we want to search on this field. Select "contains" as the search operation. Select "Param" as the search value type. Give a parameter name. Here we use "URL". Click "Add Filter" button:

Web Browsing Limit to URL List



Query builder

Query Builder - Add filters

Select field

Table: WantedPage

Field: URL (1)

Operator

<=

contains (2)

begin with

end with

Value

Field

Param (3)

Const

Name: @ URL (4)

Filters (WHERE clause)

Add filter (5)

The Query Builder generate a search statement for you:



Query builder

Query Builder - Add filters

Select field

Table: WantedPage

Field: URL

Operator

<=

<>

contains

begin with

end with

Value

Field

Param

Const

Name: @ URL

Filters (WHERE clause)

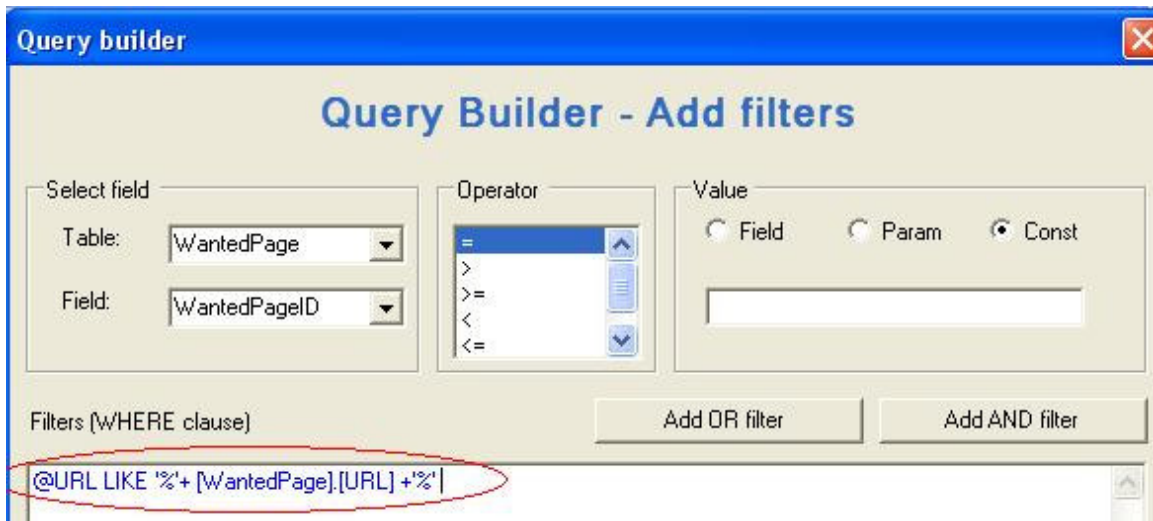
Add OR filter

Add AND filter

[WantedPage].[URL] LIKE '%+ @URL+%'

But this statement is not exactly what we want. When we use this query in our program, we will set the value of the parameter @URL to the new web page address. The statement the Query Build generated will search in such a way: the URL field in the database contains the new web page address. We want to search in the other way: the new web page address contains the URL field value in the database. For example, if the new web page address is “http://www.limnor.com/index.html”, then it contains “limnor.com” which is the value of the URL field of the first table record. To fix this problem is easy: just manually switch the field name, [Unwanted].[URL], with the parameter name, @URL:

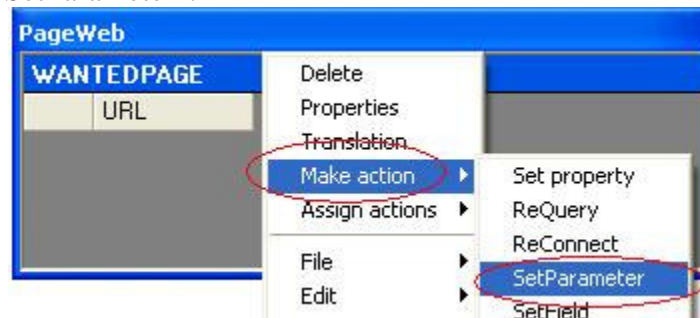
Web Browsing Limit to URL List



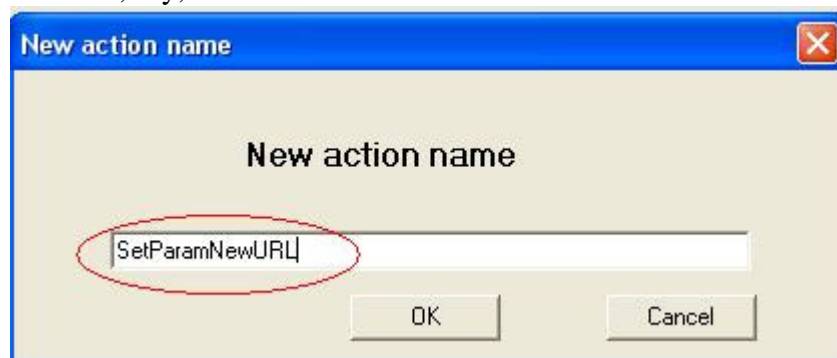
Create Database Search Actions

To search the database using the above query, we need to create two actions. One action is to set parameter @URL to the new web page address. The other action is to execute the query on the database to get result.

Create action to set parameter. Right-click on the Data Table performer, choose “Make action”, choose “SetParameter”:

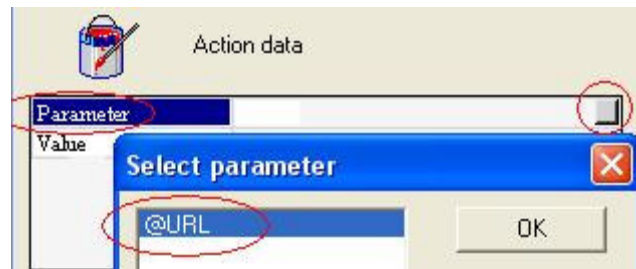


Give an action name, say, SetParamNewURL:

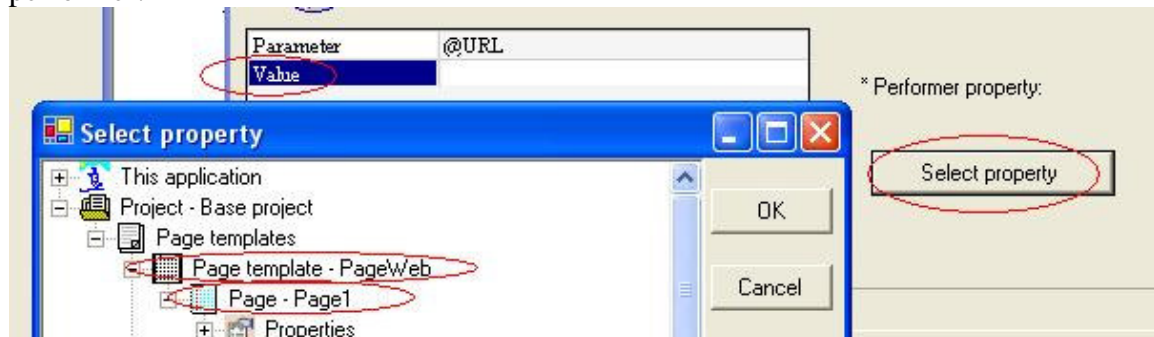


The Action Data dialogue box appears. For “Parameter”, click and choose @URL:

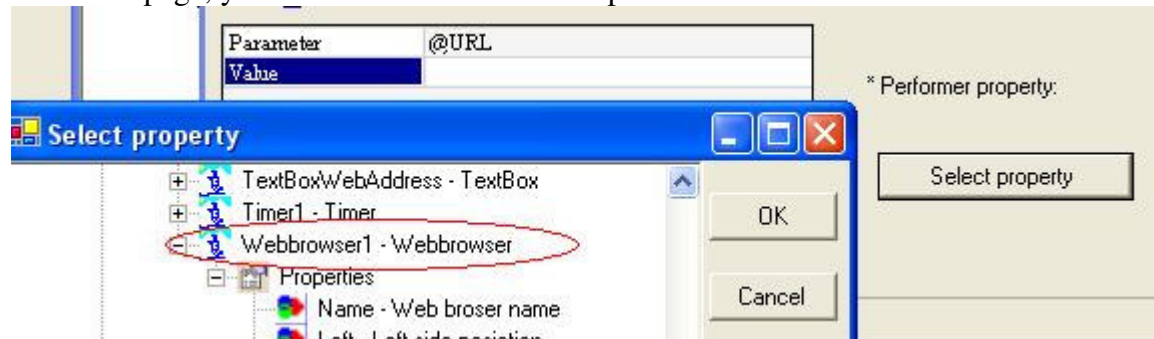
Web Browsing Limit to URL List



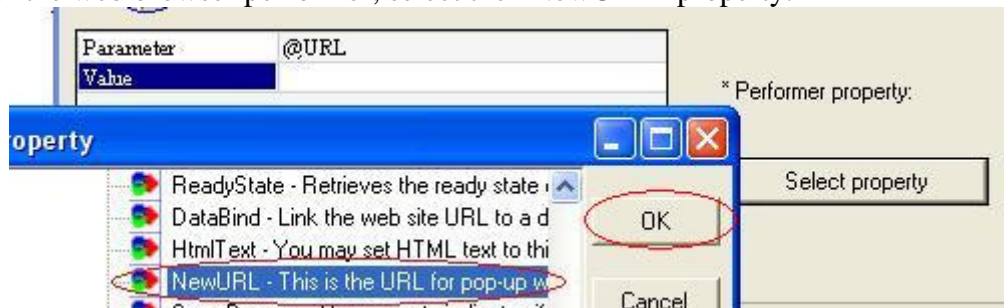
For “Value”, click “Select property” button, find the page containing the web browser performer:



Under the page, you can find the web browser performer:

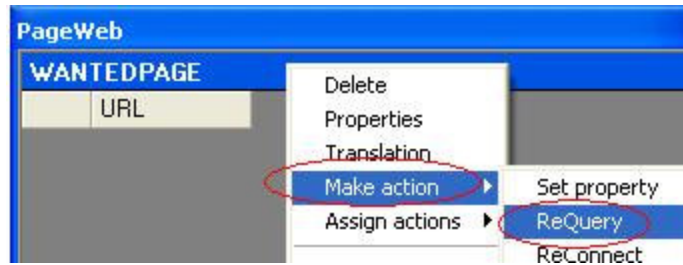


Under the web browser performer, select the “NewURL” property:

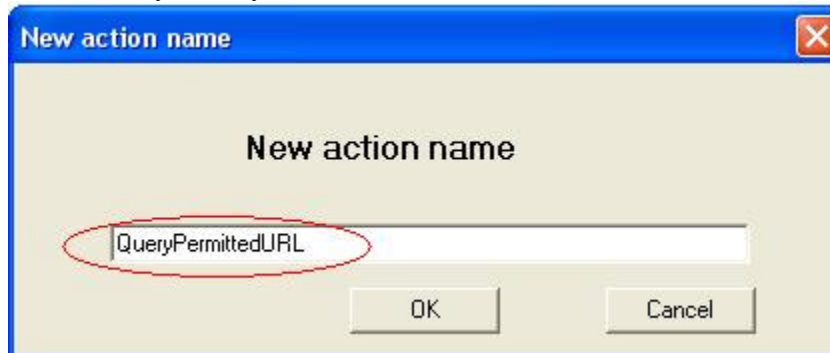


Create action to query database. Right-click on the Data Table performer, choose “Make action”, choose “ReQuery”:

Web Browsing Limit to URL List



Give an action name, say, QueryPermittedURL:

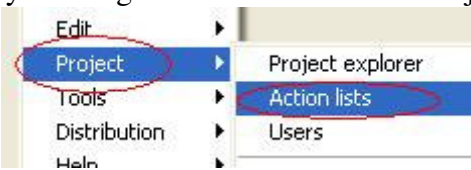


This action does not need data. So we are done.

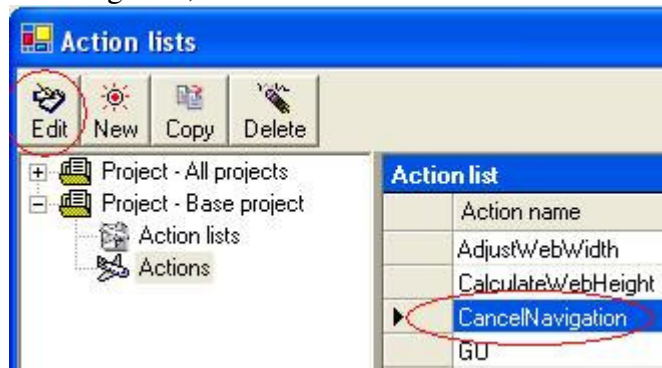
Set navigation condition

After executing action QueryPermittedURL, all records found will be listed in the Data Table performer. The Data Table performer has a “RowCount” property which indicates how many records in the Data Table. Thus, if the “RowCount” property is 0, then the new web page address is not in the white list, and should be blocked. If the “RowCount” property is greater than 0, then the new web page address is in the white list and should be allowed. So we may use the “RowCount” property to disable the action CancelNavigation.

Open “Actions” window by clicking “Actions” or choose “Project | Action lists” menu:

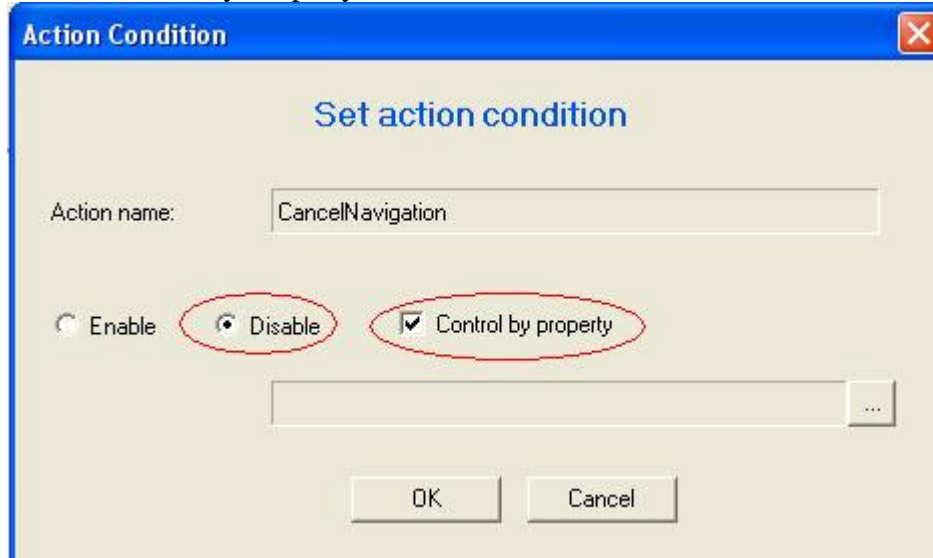


Choose action CancelNavigation, click “Edit” button:

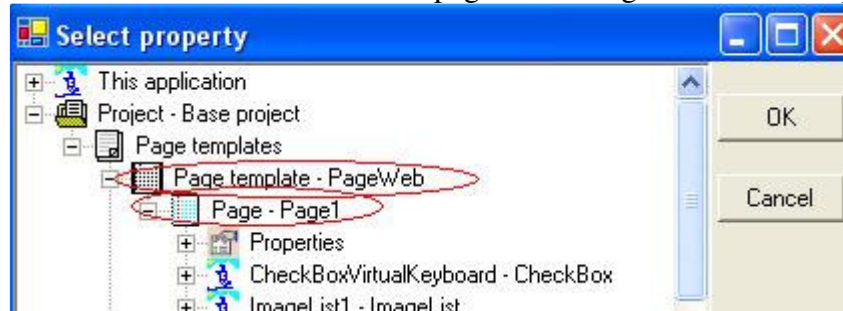


Web Browsing Limit to URL List

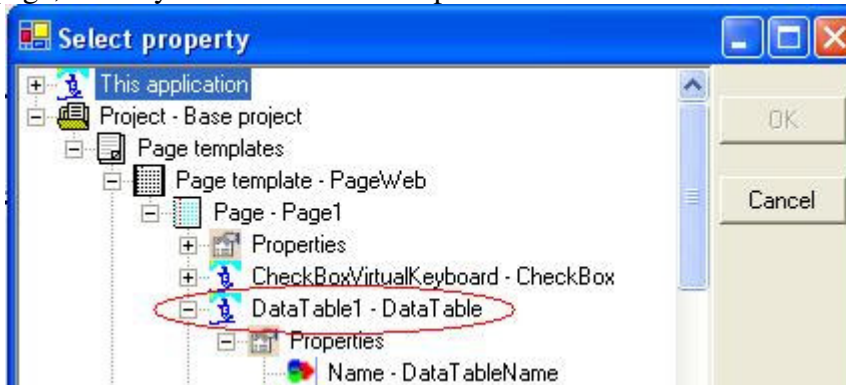
Click “Next” button. It lets you select execute condition for the action. Select “Enable” option. Check “Control by Property”:



It will ask you to select a property. We want to use “RowCount” property of the Data Table performer. So we first need to find the page containing the Data Table performer:

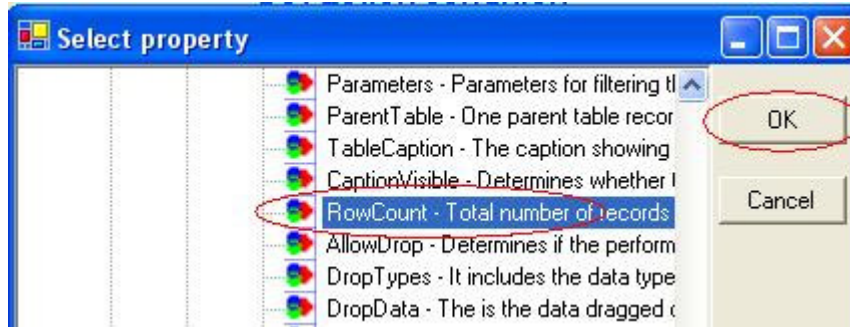


Under the page, we may find the Data Table performer:



Find its “RowCount” property, select it and click OK:

Web Browsing Limit to URL List



We are done.

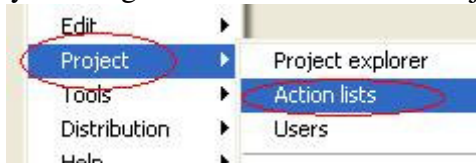
Put all actions together

By now, we have created 3 actions:

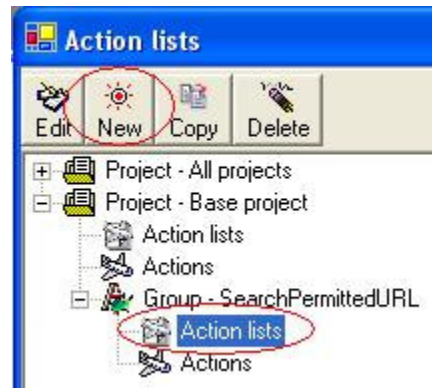
1. SetParamNewURL – Set database query parameter to the new web page address
2. QueryPermittedURL – Query database to find records from the white list, which are included in the new web page address.
3. CancelNavigation – Block web browser navigation. This action is only enabled if QueryPermittedURL action does not find records from the white list.

Now we need to create an action list to include the above 3 actions so that they will run together and in the above order.

Open “Actions” window by clicking “Actions” or choose “Project | Action lists” menu:



Select Action Lists. Click New button:

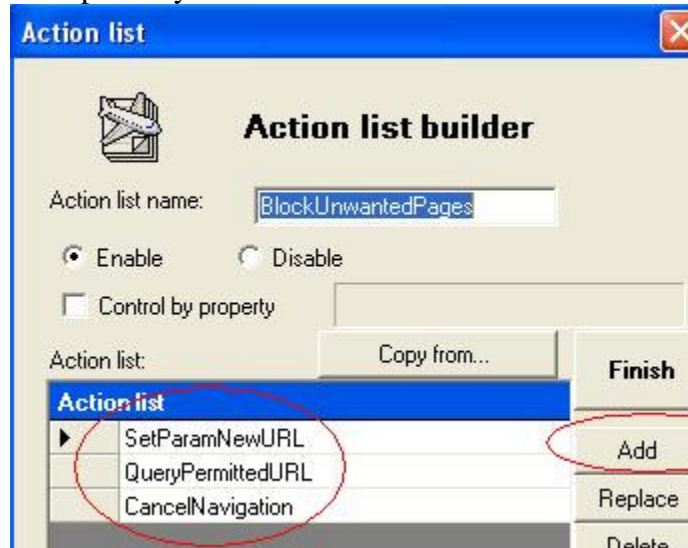


Give an action list name, say, BlockUnwantedPages:

Web Browsing Limit to URL List



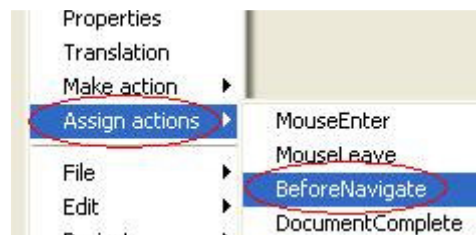
Click the Add button repeatedly to add the 3 actions to the list:



Assign Action List to Events

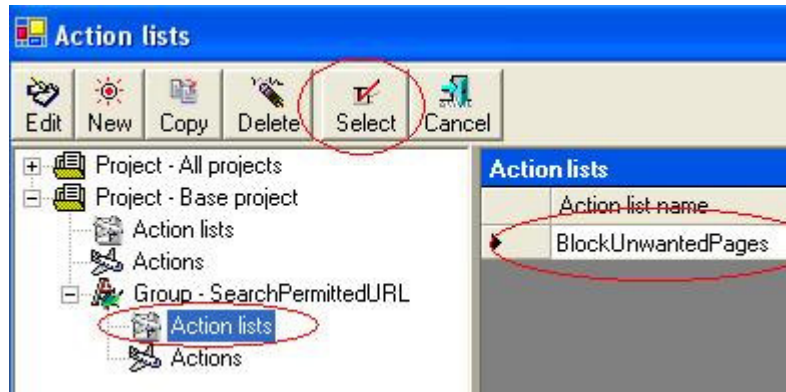
In the beginning of this article, we assign action CancelNavigation to BeforeNavigate and NewWindow events. That will block all web pages and make the web browser unusable at all. Now we have created the action list, BlockUnwantedPages, which correctly blocks the web pages we want to block. So we need to assign BlockUnwantedPages to BeforeNavigate and NewWindow events, instead of CancelNavigation.

Right-click on the web browser performer, choose “Assign actions”, choose “BeforeNavigate”:



Select “Action lists”, select BlockUnwantedPages, click “Select” button:

Web Browsing Limit to URL List



The above programming assigns action `BlockUnwantedPages` to event `BeforeNavigate`. This way, we block unwanted page navigation on the web browser. To block pop-up window, we just assign action `BlockUnwantedPages` to event `NewWindow`. Right-click on the web browser performer, choose “Assign actions”, Choose “NewWindow”:



Just as we did before, we select action `BlockUnwantedPages` for it. Now all unwanted pop-up windows will be blocked.

©2003 Longflow Enterprises Ltd. All rights reserved.