

Knowledge Base Configuration

You can download the Knowledge Base (KB) installer from <http://www.issueview.com/lvKB.exe>. You must have a copy of the IssueView database installed to use the KB.

The KB must be installed on an IIS Server, and requires the .NET Framework, version 2.0 or above. If you have upgraded from an earlier version of IssueView (5.1 or before), you'll need to make some [compatibility changes](#) to your KB Template to ensure it works smoothly with the KB web application.

You can view our KB implementation on our web site:
<http://www.issueview.com/KB/Default.aspx>

The Knowledge Base Project in IssueView

The KB draws issues from an IssueView Project, usually based on the 'Knowledge Base' Template. The Project can have any name you like. You connect the KB web site with the KB Project by editing web.config (on the IIS Server, usually in C:\inetpub\wwwroot\lvKB\web.config). Here's the key you need to edit:

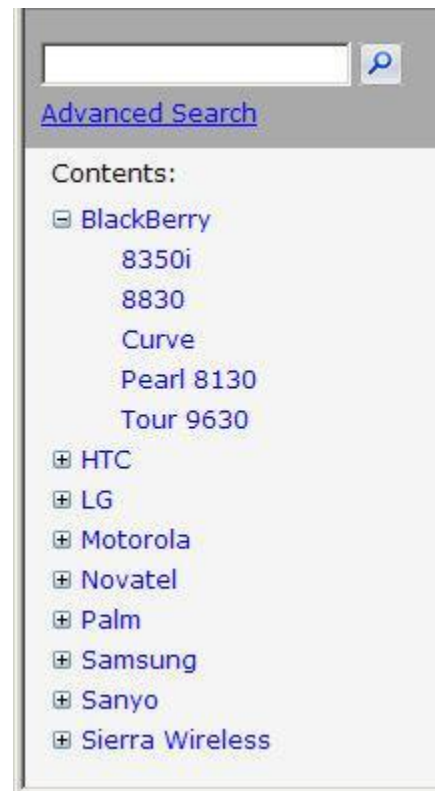
```
<add key="Project" value="Samples\Knowledge Base"/>
```

Note that if the KB Project is inside a folder, the full path information must be supplied. You can install as many instances of the lvKB as you like. You'll have to manually create the Virtual Directories in IIS. Each KB can draw from a different IssueView Project, based on the settings in web.config.

Table of Contents (Category and SubCategory)

Category and SubCategory allow you to group the data in a table of contents fashion. Eventually you'll be able to have an unlimited levels of nesting, but initially we're only supporting two levels for simplicities' sake. You may want to set up IssueView Workflow rules which limit the SubCategory choices based on the selected Category.

You can rename Category and SubCategory to anything that makes sense. We implemented the KB for a major Cellular service provider, and they track Cell Phone issues using Make and Model.



The CategoryField and SubCategoryField entries in web.config can use either the Field Name, or the actual database column name. This allows you to rename the Field in IssueView, without needed to edit web.config. But if you want to use a new IssueView Field, be sure to enter the new Field name (or db column name) in web.config

```
<add key="CategoryField" value="Category"/>
<add key="SubCategoryField" value="SubCategory"/>
```

might become

```
<add key="CategoryField" value="Make"/>
<add key="SubCategoryField" value="Model"/>
```

Article Format

The format for the data displayed in the contents pane is controlled by a file named Template.html. This file will be in the root of your KB web site on the IIS Server, usually at C:\inetpub\wwwroot\lvKB\Template.html

Before the a KB article is displayed, the article contents from the database are inserted into a dynamic copy of Template.html. You can use any html editor to make changes to Template.html, and they'll show up in the KB.

You can insert data from the selected KB article using this syntax:

```
{lvField}field name{/lvField}
```

For example, to insert the contents of the Status field into the article, add this text:

```
{lvField}Status{/lvField}
```

When the article is processed for display, the {lvField} entry will be replaced by the current value of the Status field. If the Status field for the current article is blank, no data will appear. The opening and closing {lvField} tags are case sensitive, the actual field name is not. If the Field Name has spaces (e.g., Assigned To), include them: {lvField}Assigned To{/lvField}

Customizing the Header

The contents of the KB are displayed in an HTML frameset contained in a file named Default.aspx. The top frame, displays the contents of Header.aspx. By editing Header.aspx, you can integrate the Knowledge base into your web site.

Here's the default header from the initial installation



And here's the customized header we use on our web site:



Search Fields

By default, the searches will look for the search text in all the Text Tab fields, plus the Title Field (which is the Abstract field by default). If you wish to search different fields, you can specify them in web.config. If you specify Fields in web.config, ONLY those fields will be searched:

```
<IssueViewKB>
<SearchColumns>
<add key="Abstract" value=""/>
<add key="Description" value=""/>
</SearchColumns>
</IssueViewKB>
```

Article Sort Order

By default, articles returned from searches are sorted by the IssueView ID, in Descending order. If you wish to sort by a different Field, you can specify the Order By clause in web.config. You must use the actual database column name for the field.

This example sorts by the Primary Key, in Ascending order:

```
<add key="OrderBy" value="Order By Issue ASC"/>
```

Copying links

The default article template contains a button that will copy an html link to the current article to the clipboard. This is extremely useful for pointing others to specific articles in your KB. In order for this functionality to work, IvKB has to know what your Base URL

is. This can be set in web.config. Here's the setting for our production server. When setting up your KB, enter your own URL:

```
<add key="BaseURL" value="http://www.IssueView.com/lvKB"/>
```

Filtering Articles

You can control which issues are available for searching with Filter SQL. This web.config entry requires valid SQL (if you need help, drop us a line: support@issueview.com).

Here's an example that will display articles whose status is either 'Published', or 'For Review'

```
<add key="FilterSQL" value="Status = 'Published' or Status = 'For Review' "/>
```

Compatibility Changes (pre 6.0 databases)

We changed the IssueView 'Knowledge Base' Template in version 6.0 to support our new KB web application. If you have upgraded from IssueView 5.1 or before, you should make the following changes to your 'Knowledge Base' Template:

1. In the original KB Template (pre - 6.0) we had Fields named Submitter, Type, Subject and Description. Those fields have been replaced with new Fields

- Rename Contacts field Submitter to be Author.
- Rename Choice field Type to be Category
- Rename Choice field Subject to be SubCategory
- Rename Text Tab field Description to be Article

2. Add a Choice field named Status. Add the choice values Draft, For Review, and Published.

OR

If you have no data in your Knowledge Base, just delete the Submitter, Type and Subject Fields, and create the new fields named Author, Category, SubCategory, Status, and Article. This will ensure the Field names match the default entries in web.config.

Custom Search Links

The simplest way to search the Knowledge Bank via a link is by encoding Search Terms in the QueryString. To search the knowledge base for articles containing the phrase 'setup issues', you could use this URL:

<http://www.mycompany.com/IvKB/Default.aspx?Search=setup%20issues>

Note the use of URL encoding to represent a space with %20. QueryString based searches will match articles containing All Words (rather than Any Words or an Exact Phrase)

Custom Search Forms

You can design a custom HTML form and use a Form POST to execute a search. A sample form named FormSearch.htm is included with your IvKB site. Here are the available form fields:

```
<form name="form1"
  method="post"
  action="http://localhost/IvKB/LeftSearch.aspx">
<input name="CategoryDBCol" type="hidden" value="Category" />
<input name="SubCategoryDBCol" type="hidden" value="SubCategory" />
<input name="SearchString" type="text" value="search text" />
<select name="SearchType">
  <option value=0>Exact Phrase</option>
  <option value=1 selected="selected">Match All Words</option>
  <option value=2>Match Any Words</option>
</select>
<select name="SearchPeriod">
  <option value=0 selected="selected">Any Time</option>
  <option value=1>One Day</option>
  <option value=2>One Week</option>
  <option value=3>One Month</option>
  <option value=4>Six Months</option>
  <option value=5>One Year</option>
</select>
<select name="CategoryValue ">
  <option value=" " selected="selected">&nbsp;</option>
  <option value="Category Value 1">Category Value 1</option>
  <option value="Category Value 2">Category Value 2</option>
  <option value="Category Value 3">Category Value 3</option>
</select>
<select name="SubCategoryValue">
  <option value=" " selected="selected">&nbsp;</option>
  <option value="Sub-Category 1">Sub-Category 1</option>
  <option value="Sub-Category 2">Sub-Category 2</option>
  <option value="Sub-Category 3">Sub-Category 3</option>
</select>
<input type="submit" value="submit" />
</form>
```

You can give the form any name you want. Be sure to edit the 'action' URL. The example uses 'localhost', which will work if you're testing on the IIS Server; but you'll need to enter a real URL for other users to reach the site.

The CategoryDBCol and SubCategoryDBCol hidden fields allow you to

customize the search. Note that they are the actual column names in the table, not the IssueView field names (though they are often the same). You can omit these fields.

Searches use the 'Match All Words' setting by default, even if you omit this field.

There are no more elaborate options for SearchPeriod. It is not possible to enter custom date ranges.

CategoryValue and SubCategoryValue correspond to CategoryDBCol and SubCategoryDBCol. Normally the person using your form will only be entering a value; the available Columns are fixed and known when you design the form.