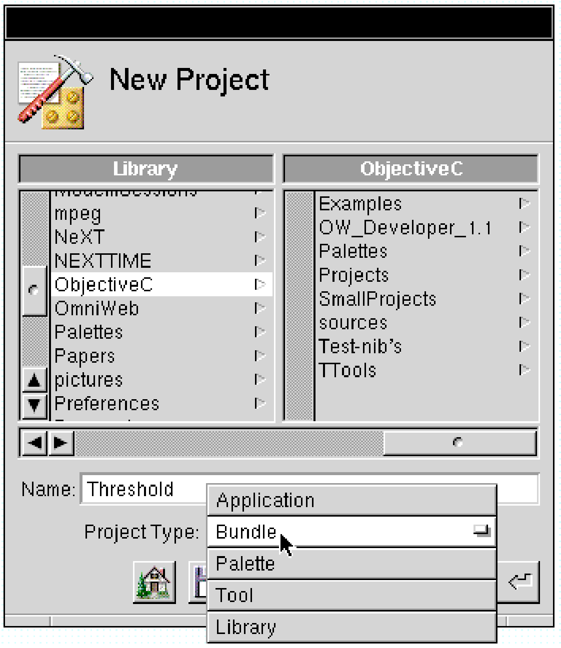
Implementing Image Operators

The ACRViewer application is extensible. Any global image processing algorithm can be implemented within a bundle project that will be dynamically loaded by ACRViewer. This paper demonstrates step-by-step with an example implementation of the threshold operator, how to implement such an image operator and how to integrate it into the ACRViewer application.

Creating the Project

Launch ProjectBuilder and choose "Project/New..." from ProjectBuilder's main menu. In the appearing file selection panel, choose "Bundle" as project type.

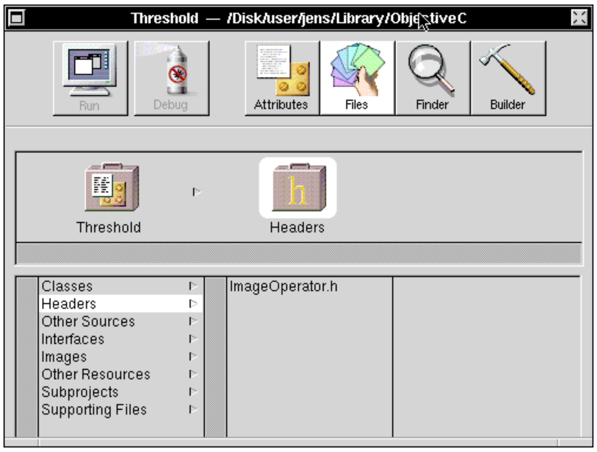


After entering the project name (Threshold) and clicking the OK button, ProjectBuilder will generate the needed Makefiles but there won't be anything else in the project at this time.

Note: The source code of the threshold operator and the interface file come with ACRViewer.

Any image operator extending ACRViewer has to be a subclass of the ImageOperator class declared in the ImageOperator.h file that comes with ACRViewer. This class has to be known by the project. This can be achieved by either editing the header include path in Makefile.preamble, or dragging the ImageOperator.h file into the Headers suit-case of ProjectBuilder.

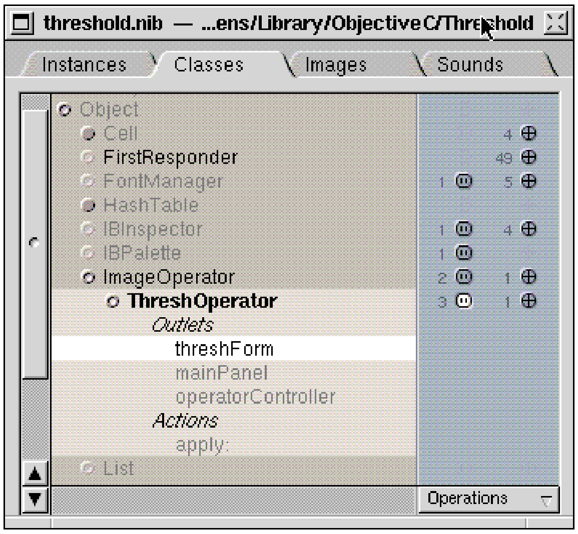
In the screen shot below this file has simply been dragged into the Project:



ProjectBuilder does not generate a standard interface file for bundle projects. We will handle this task in the next section.

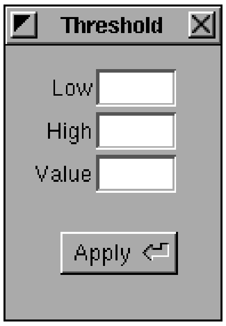
Creating the Interface

In InterfaceBuilder (IB) choose "Document /New Module/New Empty". This will bring up an IB document with only two objects: The "File’s Owner" and the "First Responder".
Before we continue, save the new IB document in the Threshold project directory with "threshold.nib" as filename. IB will offer you to insert the interface into the Project. We want this, so click YES.
Now we want to create our subclass of ImageOperator (the ThreshOperator class). Before we can do so, IB must be aware of the ImageOperator class. Switch to the classes display in IB and choose "Parse" in the "Operations" pull-down menu. Select the "ImageOperator.h" file you’ve just inserted into the Threshold project. Now that IB displays the ImageOperator class, select it and choose "Subclass" from the "Operations" menu. Name the subclass ThreshOperator. Add an outlet to the ThreshOperator class and name it threshForm.



"Unparse" the ThreshOperator class and let IB insert the ThreshOperaror.[hm] files into the Project.

We won’t create an instance of the ThreshOperator directly. Instead we will use the "File’s Owner" object in the instance display: Change the class of the "File’s Owner" to ThreshOperator.



Create a Panel object that looks like the one above and select "Hide on deactivate" in the panel’s attributes inspector. Now, connect

- the mainPanel outlet of "File’s Owner" to the Threshold panel,
- the threshForm outlet of "File’s Owner" to the Form object in the Threshold panel,
- the target of the "Apply" button to the "File’s Owner" with apply: as action method,
- the target of the Form object in the Threshold panel to the "Apply" button with performClick: as action method.

Save the interface document.

Implementing the Operator

Open the ThreshOperator.m file in ProjectBuilder and add the following source into the generated source code frame:

```
- (const char*) operatorName
{
  return "Threshold";
}

- openPanel
{
  [self loadInterface:"threshold.nib"];
  return [super openPanel];
}

- processImage:(unsigned short*) data
  withWidth:(int)width
  height:(int) height
  minValue:(unsigned short) minValue
  andMaxValue:(unsigned short) maxValue
{
unsigned int low= [threshForm
intValueAt:0];
unsigned int high= [threshForm
intValueAt:1];
unsigned int inValue= [threshForm
intValueAt:2];
unsigned int v;
int i;
int count = width*height;

  if (inValue == 0)
    for (i= 0; i<count; i++){
      v= data[i];
      if (v < low || v>high) v= 0;
      data[i]= v;
    }
  else
    for (i= 0; i<count; i++){
      v= data[i];
      if (v < low || v>high)
        v= 0; else v= (int)inValue;
      data[i]= v;
    }
  return self;
}
```

For more information read the ImageOperator.h file.

Save the source code and build the project in ProjectBuilder with target "bundle". There shouldn't occur any errors.

Installing the Operator

ACRViewer loads all bundles that are in the app wrapper. Another way to get ACRViewer loading your bundle is Command-dragging a bundle on the application icon.
So, to use the ThreshOperator in ACRViewer, you can either install the ThreshOperator in a path like ~/Library/ACRViewer (and load it explicitly by Command-dragging the bundle icon on the application icon) or you can install it in the app wrapper. In ProjectBuilders "Attributes" display change the installation path to whatever you prefer and build the project with target "install".