

# The Ictetips Previewer



Your own customizable Clarion Report Previewer

## Table of contents

|                                  |    |
|----------------------------------|----|
| Table of contents                | 2  |
| Welcome                          | 6  |
| Introduction                     | 7  |
| Installation                     | 8  |
| Installed components             | 8  |
| Quick Start Tutorial             | 9  |
| Preparing                        | 9  |
| Using the Wizard                 | 10 |
| Adding the Previewer to a report | 12 |
| Options when calling Previewer   | 14 |
| Clarion 6 options                | 15 |
| Done and ready to go             | 16 |
| The Previewer Window             | 17 |
| Toolbar                          | 17 |
| Pages to Print                   | 17 |
| Search                           | 18 |
| Copies                           | 18 |
| Zoom, Step                       | 18 |
| Toggle / PageList                | 18 |
| Full source provided             | 19 |
| Previewer code and design        | 19 |
| Prototype and Parameters         | 21 |
| Changing the Previewer           | 22 |
| Removing functionality           | 22 |
| Adding functionality             | 23 |
| Wizard template                  | 24 |
| Run the Wizard                   | 24 |
| Enter Procedure name             | 24 |
| Visual theme selection           | 25 |
| Selecting a Theme file           | 25 |
| Clarion 6 Options                | 26 |
| Clarion 6 compatibility          | 26 |
| Fully customizable               | 26 |
| Extension Templates              | 27 |
| Global Extension                 | 28 |

|   |           |
|---|-----------|
| Enable Printer Selection                              | 28        |
| Include PRNPROP.CLW                                   | 28        |
| Prototype ShellExecute for C6 target classes          | 28        |
| Generate no code                                      | 29        |
| Previewer Extensions                                  | 30        |
| Icetips Previewer - Enable printer selection          | 30        |
| Printer dialog header                                 | 30        |
| Variable for header                                   | 30        |
| Local page Queue                                      | 31        |
| Page variable   | 31        |
| Queue parameter                                       | 31        |
| Free Queues   | 31        |
| Select Printer first                                  | 31        |
| Generate No code                                      | 31        |
| New Local data  | 31        |
| New Routines  | 31        |
| ITPPrintToSelectedPrinter                             | 31        |
| ITPPrintCurrentPageToSelectedPrinter                  | 31        |
| ITPSelectPrinter                                      | 32        |
| How to implement printer selection                    | 32        |
| Icetips Previewer - Support for Clarion 6             | 32        |
| How to implement Clarion 6 support and export options | 32        |
| Report Extension                                      | 35        |
| Apply the Extension                                   | 35        |
| Select Extension template                             | 35        |
| Icetips Previewer Tab                                 | 37        |
| Zoom type   | 37        |
| Start Maximized                                       | 37        |
| Start Pagelist  | 37        |
| Previewer Caption                                     | 37        |
| Code generation options                               | 37        |
| Condition for use                                     | 37        |
| Do NOT call previewer                                 | 37        |
| Allow additional parameters                           | 37        |
| Additional parameters                                 | 38        |
| Generate No code                                      | 38        |
| Information tab                                       | 38        |
| Previewer Extension Embed points                      | 38        |
| Note about version 1.001                              | 40        |
| Note about version 2.000                              | 40        |
| <b>Control Templates</b>                              | <b>41</b> |
| Icetips Previewer PDF Tools button                    | 41        |
| Adding the control template                           | 41        |
| New embeds  | 41        |
| Generated code  | 42        |

|   |           |
|---|-----------|
| Save buttons for Clarion 6  | 43        |
| Retrieve target filename  | 44        |
| Open target file in associated program                                | 44        |
| Generate No code  | 44        |
| Generated code  | 44        |
| OpenWindow Event Handling:  | 44        |
| Control Event Handling, ?Clarion6Button1 - ?Clarion6Button4, Accepted | 44        |
| Procedure Routines:   | 44        |
| Save As button for Clarion 6  | 47        |
| Retrieve target filename  | 48        |
| Open target file in associated program                                | 48        |
| Generate No code  | 48        |
| Generated code  | 48        |
| Control Event Handling, ?C6SaveAsButton, Accepted                     | 48        |
| Procedure Routines:   | 48        |
| <b>What is new in version 2.0</b>                                     | <b>51</b> |
| New templates   | 51        |
| Fixes and updates   | 51        |
| <b>The Icetips Previewer Editor</b>                                   | <b>53</b> |
| Main window   | 54        |
| The Editor window   | 55        |
| Toolbar Buttons tab   | 55        |
| Filenames only for icons option                                       | 56        |
| Icon Paths as Tooltips  | 56        |
| Search Path button  | 56        |
| View Previewer button   | 57        |
| Toolbar Buttons and Window tab  | 57        |
| Flat Buttons option   | 57        |
| Previewer Window and pagelist background                              | 57        |
| Previewer Window statusbar  | 57        |
| Cursor  | 58        |
| Theme option  | 58        |
| Save As   | 58        |
| Tooltips + Prompts and Texts  | 59        |
| The Search Path window  | 60        |
| The .itp file format  | 60        |
| <b>Compatibility &amp; technological issues</b>                       | <b>62</b> |
| Compatibility with earlier versions of Clarion                        | 62        |
| Clarion 4, ABC  | 62        |
| Optionally using the Clarion or CPCS standard previewers              | 63        |
| Clarion 5, Legacy   | 63        |
| Other Clarion third party products                                    | 63        |

|   |           |
|---|-----------|
| <b>Demos</b>  | <b>65</b> |
| Old demos   | 65        |
| New demos   | 65        |
| Clarion 6 demo  | 65        |
| Adding target support to your Clarion 6 application and Icetips Previewer | 65        |
| Registering the templates   | 66        |
| Adding Target templates to your application                               | 66        |
| Add the Icetips Previewer Global Extension template                       | 66        |
| Add Clarion 6 support to the Icetips Previewer procedure                  | 66        |
| Modify the Icetips Previewer Prototype                                    | 67        |
| Add Target buttons to the Icetips Previewer                               | 68        |
| Set Target button options   | 69        |
| Compile and run   | 69        |
| <b>Installed files</b>  | <b>70</b> |
| Files in: %ROOT%\3rdParty\Docs\ITPrev                                     | 70        |
| Files in: %ROOT%\3rdParty\Examples\ITPrev                                 | 70        |
| Files in: %ROOT%\3rdParty\Images  | 70        |
| Files in: %ROOT%\3rdParty\Template  | 72        |
| Files in: %ROOT%\3rdParty\Tools\ITInstall                                 | 72        |
| Files in: %ROOT%\3rdParty\Tools\ITPrev                                    | 72        |
| Files in: %ROOT%\Bin  | 72        |
| <b>Technical Support</b>  | <b>73</b> |
| Email   | 73        |
| On the web  | 73        |
| Icetips Software  | 73        |
| Icetips Bulletin Board  | 73        |
| Icetips Software Online Support   | 73        |
| Newsgroups  | 73        |
| Mail  | 73        |

## Welcome

Welcome to the Icetips Previewer version 2.0. This product, like so many others, was created to meet our own needs: a report previewer that could be customized based on client needs. It also allows the developer to change the look and feel of the visual controls to conform with their own visual standards.

Since the Icetips Previewer was released in January 2002, it has become one of our most popular products because of its flexibility and ease of use. It is currently being used by hundreds of Clarion developers and thousands of users around the world.

With version 2.0 we have added support for the new Clarion 6 report target export classes which enable you to export the report to PDF, HTML, TeXe or XML directly from the previewer. It also adds options to print the report to a different printer.

## Introduction

The Icetips Previewer is created as a window procedure in your application. All the code necessary for the Previewer's functionality is in embedded code in the previewer procedure. The developer can modify any part of it, and remove or add functionality if needed. You can create multiple previewers for one application and use them for separate reports, or call each one conditionally, based on user login for example.

To make calling the previewer easier, we provide an extension template that you apply to the reports. Reports that you don't apply the extension template to will simply call whatever previewer that is the standard.

The Icetips Previewer is compatible with the CPCS reporting templates as well as both ABC and Legacy (Clarion) templates. The Icetips Previewer is compatible with and has been tested with Clarion 4, Clarion 5, Clarion 5.5 and Clarion 6.0

In this document we will go through the Installation, the Previewer and the Wizard and Extension templates that generate the Previewer and add it to your reports.



*Tips and  
info*

Note that where you see the light bulb image in the left margin, it means that there is some important tips or information in the text.

Also note that in some places we have edited windows a little bit to make them fit into this document - made them narrower than they really are, or shorter than they really can be in the Clarion IDE. When you see screenshots in this document that don't look "real" that is because we have "doctored" them a little bit to make them fit.

*In some places we have added some technical notes. These notes are printed in a different font and have a frame around them.*

## Installation

The Icetips Previewer is supplied as a standalone ITPreview.exe installation file created with SetupBuilder 4 from Linder Software. During the installation, the install program should register the templates in the Clarion version you selected to install it for.

After the installation is complete, make sure that the template was registered without problems.

Now you are ready to use the Icetips Previewer Wizard in your applications to create your own previewer and then apply the Extension template to reports that you want to use your new previewer. The Wizard template and the Extension templates are discussed later in this document. You can at any time go into your Previewer procedure and make any changes you want to. The Wizard template creates the Previewer procedure either by using predefined color themes for the icons, or by using your own, which you set up in the Icetips Previewer Editor. Later in this document we briefly discuss the internals of the Previewer.

## Installed components

The installation installs several components for you

1. The Wizard template
2. The Documentation in PDF format
3. The Previewer Editor
4. Supporting programs that update your redirection file, etc.

Now, let's take a look at the Previewer procedure the wizard will create for you and the functionality that is provided as a standard.

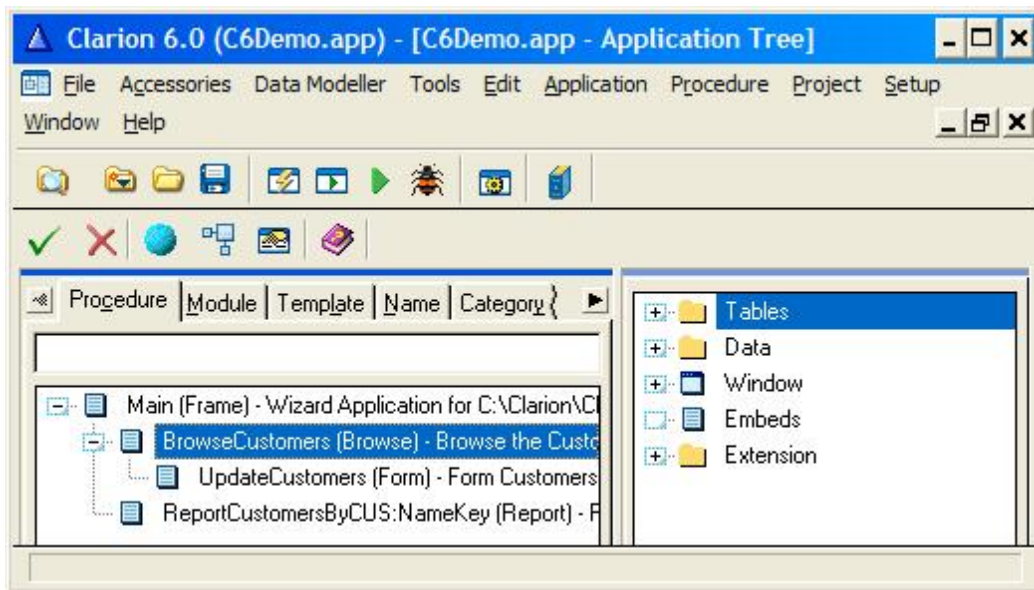


## Quick Start Tutorial

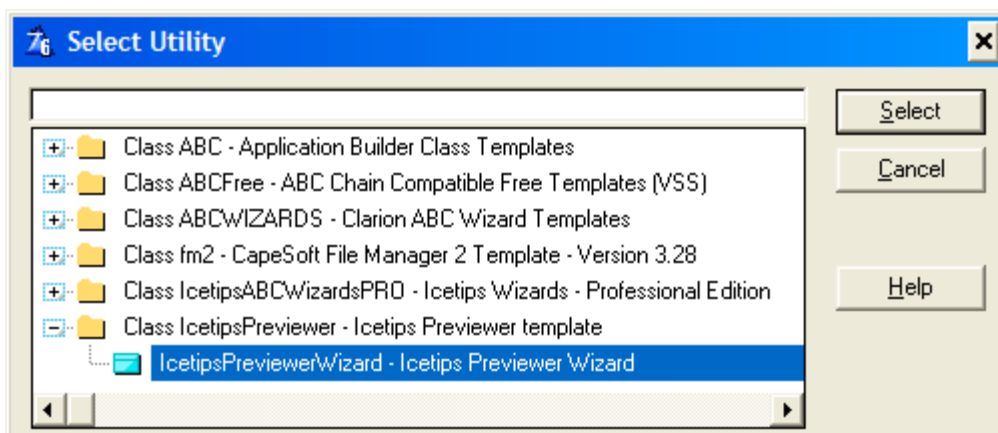
This section gives a quick reference to what you need to do to get the Icetips Previewer up and running in your application once you have installed the product and made sure the templates are registered.

### Preparing

1. Start Clarion.
2. Open the application you are going to add the Previewer to.



3. Select "Application|Template Utility" from the Clarion main menu.



4. Locate the "IcetipsPreviewer" class in the list of available utility template classes.
5. Select the "IcetipsPreviewerWizard - Icetips Previewer Wizard"
6. Click the "Select" button on the "Select Utility" window.

- The Wizard will now run and prompt you for some information and then create the previewer procedure.

## Using the Wizard

- Enter the name of the procedure you want to create. You can use any name you want, but it must be a valid Clarion procedure label. I.e. it can not have spaces or any other characters that would make it an invalid label. IcetipsPreviewer or ITPreviewer are just fine. We will use IcetipsPreviewer for this exercise. When you have entered the name, click on the "Next" button to go to the next window.



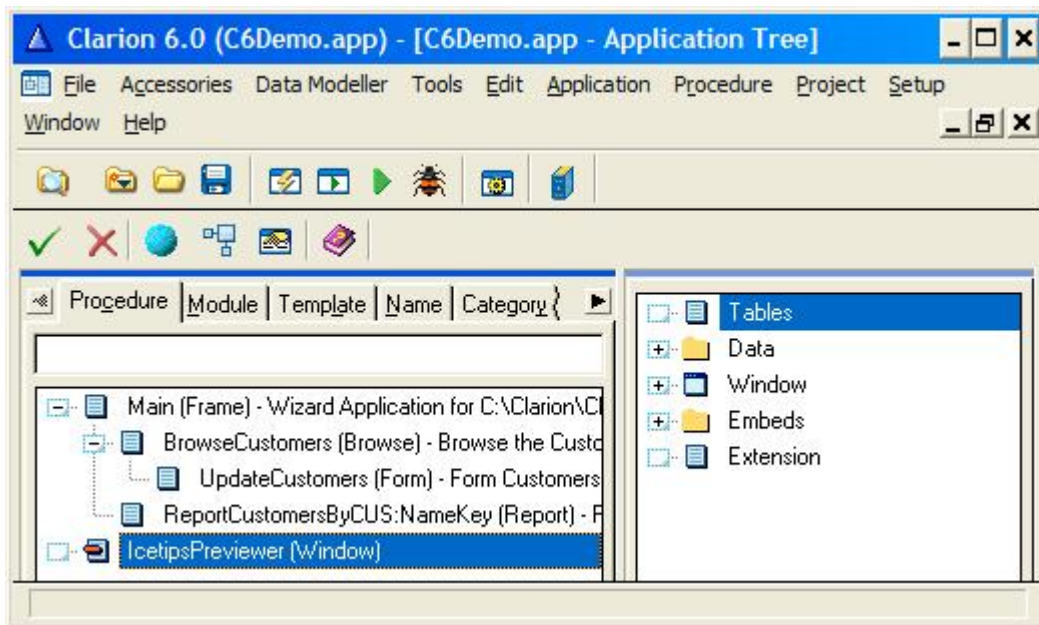
- Next window allows you to select a color theme for the previewer. The previewer comes with 5 color themes that applies to all the icons that are used in the window. You can also create your own themes using the Previewer Editor and then select it by selecting the "From file" option. For this previewer, we are going to use the Blue theme so we select "Blue" from the option list.



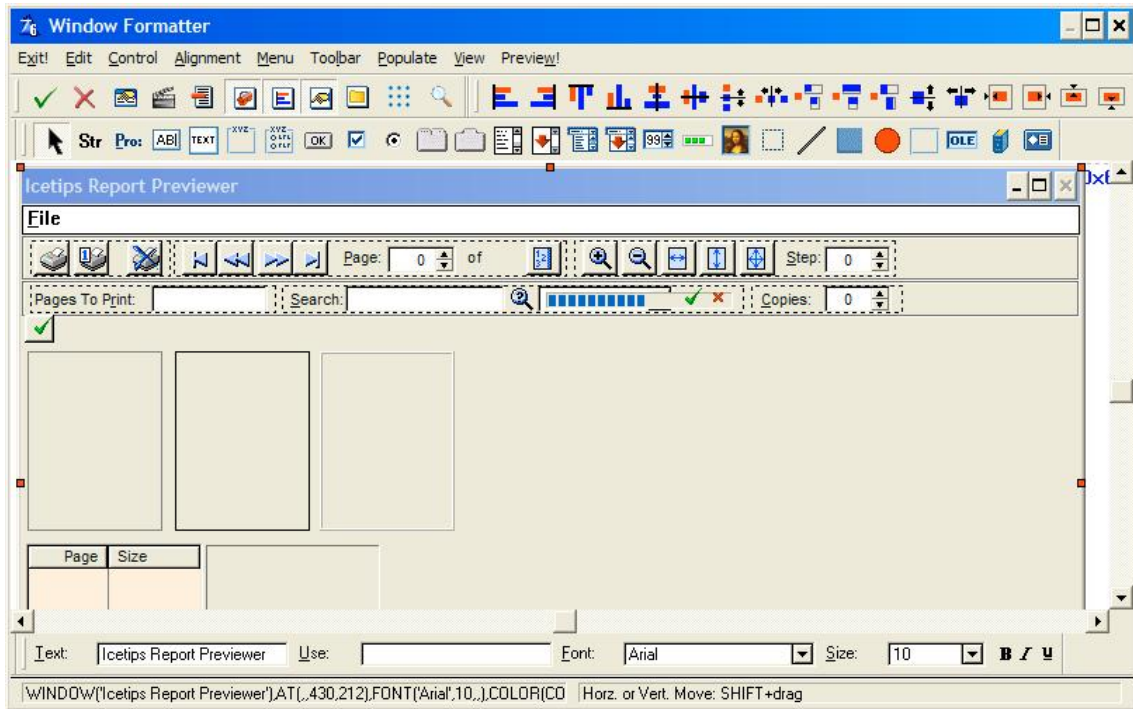
- The final option in the wizard is only available if you are running the wizard in Clarion 6. It will not show up if you are running the wizard in Clarion 4, Clarion 5 or Clarion 5.5. The only option on this window, "Generate Target parameter" will automatically add one extra parameter to the previewer. This parameter is omittable so it will not break any existing applications that are already using previous versions of the Icetips Previewer. This parameter is used when implementing the Clarion 6 export target classes and passes the main target class from the report procedure to the previewer. This enables the previewer to access all the information as it was set up in the report and use it when generating the export files. Note that if you check this option, you MUST add the Clarion6Support template to the previewer procedure when it has been created.



- Click the Finish button on the last screen in the Icetips Previewer Wizard. It will now generate the IcetipsPreviewer procedure in your application.



- If you double click on the "Window" tree node in the right hand pane in the Clarion IDE (or double click the procedure to bring up the Procedure properties window, then click the "Window" button) you will see the new previewer window in the window formatter.

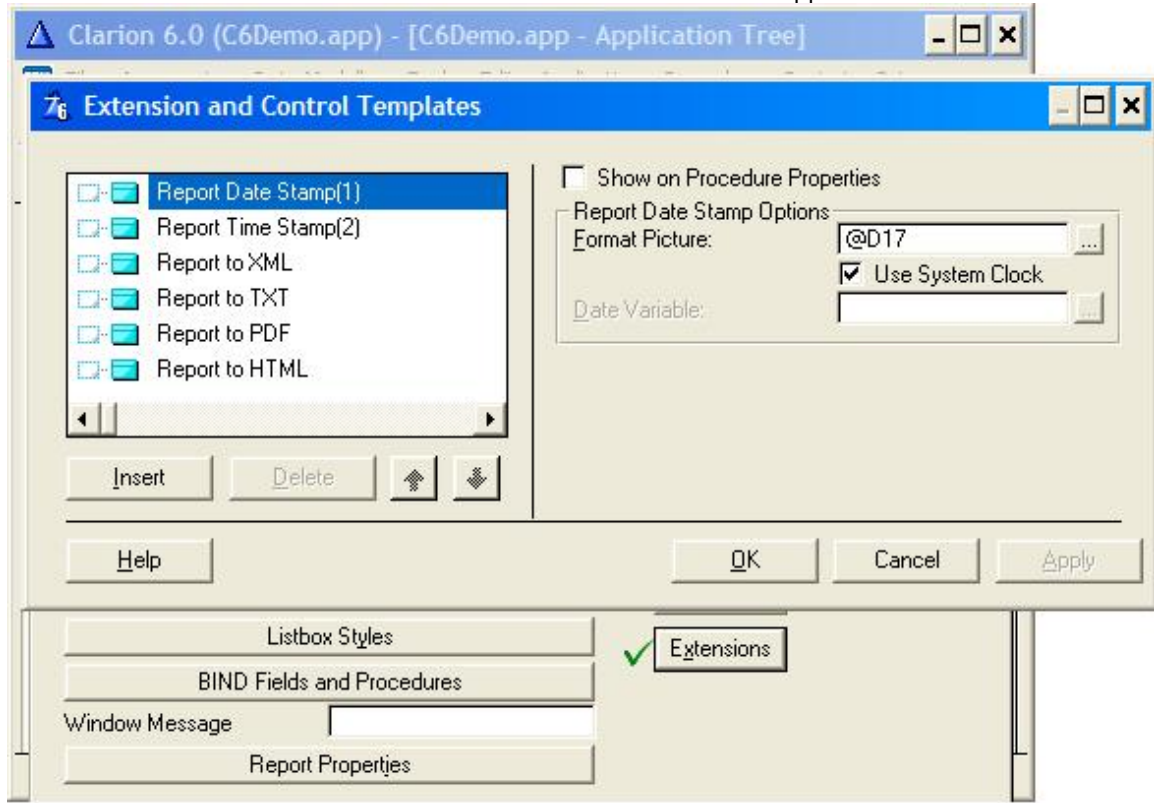


- Go back to the application tree view.

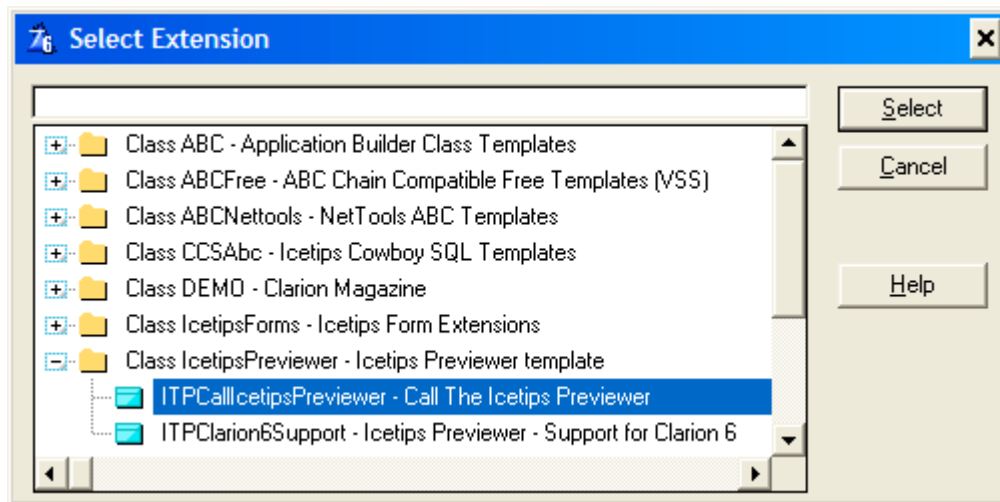
## Adding the Previewer to a report

- Double click on a report procedure that you want to add the previewer to.

15. Click on the “Extensions” button. You will now see a list of the applied extensions.



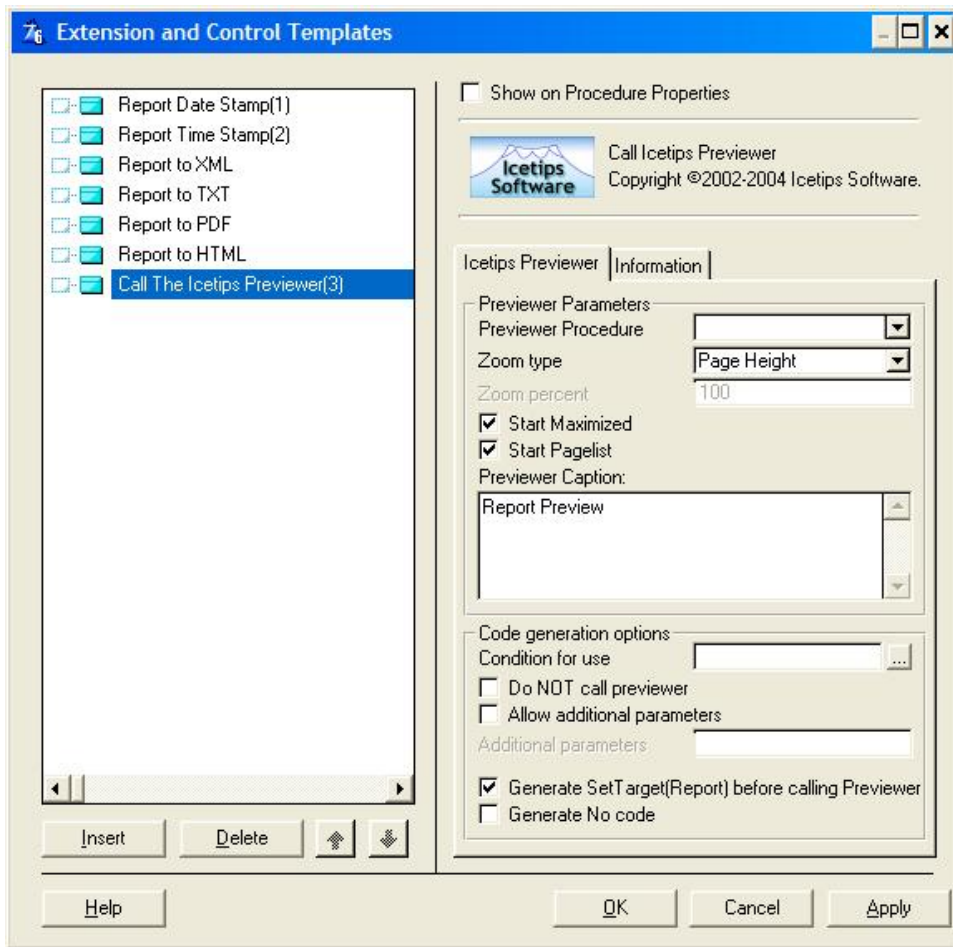
16. Click on the “Insert” button to add the Previewer extension template.



Locate the “Icetips Previewer” class and select the “ITPCallIcetipsPreviewer - Call The Icetips Previewer” template from the list.

17. Click on the Select button.



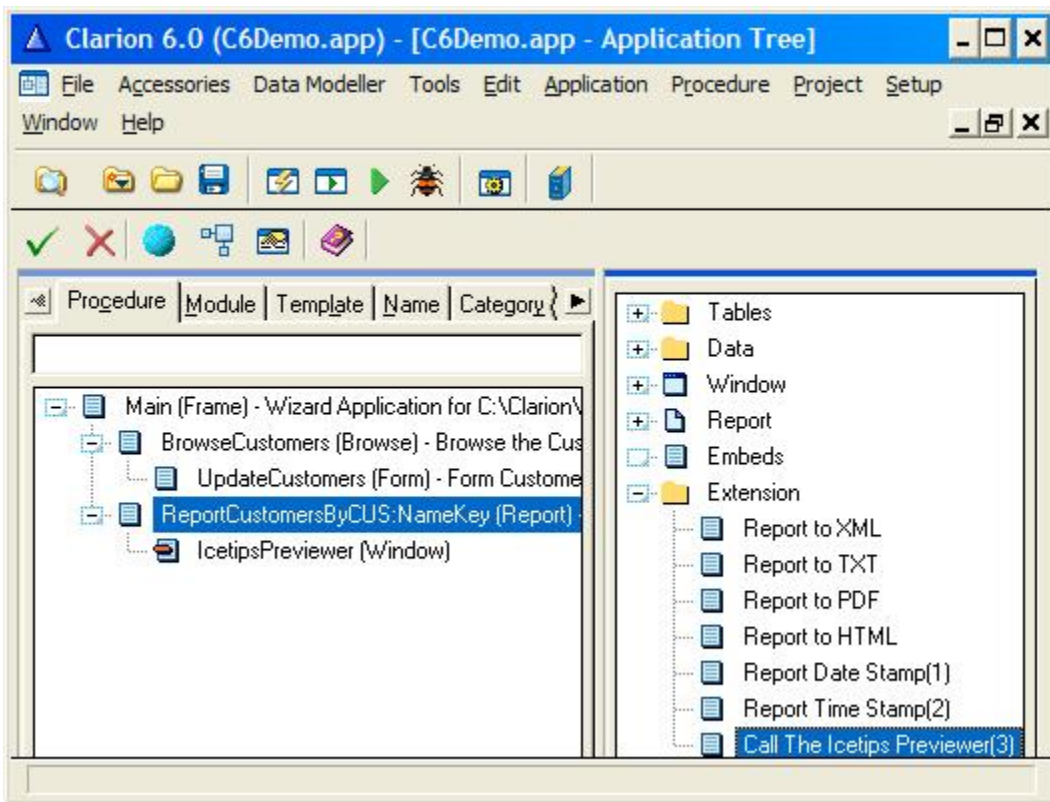


18. The Icetips Previewer extension is now added to the report.

## Options when calling Previewer

19. In the "Previewer Procedure" dropdown, select the Previewer procedure you created earlier (created in step 11)
20. Optionally select the appropriate zoom values from Page Height, Page Width, Full and Value. Value gives you an option to set the zoom value using the Zoom Percent entry field.
21. Check if you want the previewer window to start maximized (default) and if you want the pagelist listbox to show (default on).
22. Change the report previewer caption if you want to.
23. Set conditional use if you want to. That means that the previewer will only be called if the condition evaluates to true. This condition is created in a simple IF statement and must therefor obey the rules of an IF statement.
24. If you want to pass more information to the previewer, you can add parameters. You must then change the prototype of the Previewer procedure accordingly.
25. Check the Generate SetTarget() if you need to have a reference to the report in the previewer. This can be useful if you for example want to check settings such as print job name etc. in the previewer.
26. If you want to disable the code generation, for example while debugging a report, check the "Generate No code" and no code will be generated by the template.
27. Click OK to close the Extension window.

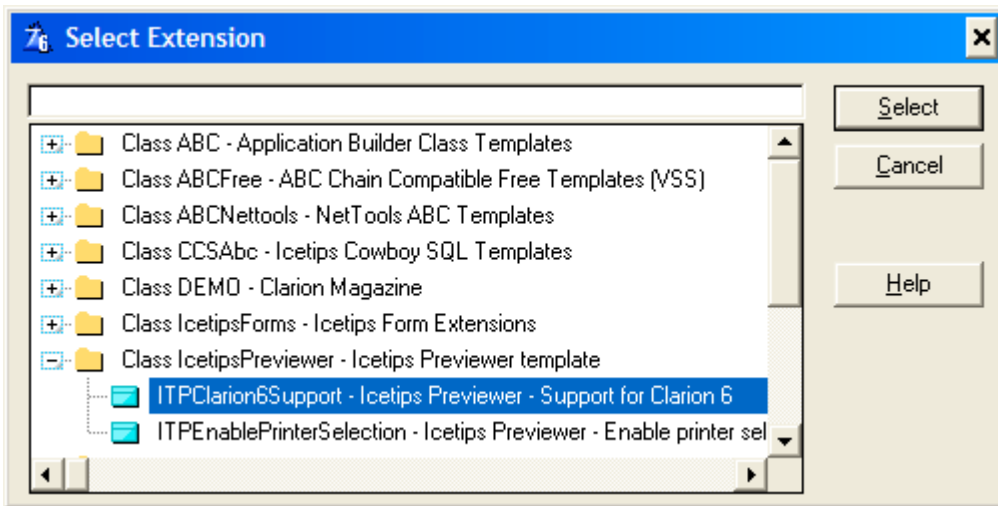
28. The Icetips previewer procedure, IcetipsPreviewer, which you created is now attached to the report.



29. If you are running Clarion 6 and you checked the "Generate Target parameter" when you created the Previewer using the wizard, then you need to follow the additional steps. If you do not do that, you will get compile errors on the previewer procedure. If you are not using Clarion 6, then you can skip to step 40 on page 16.

## Clarion 6 options

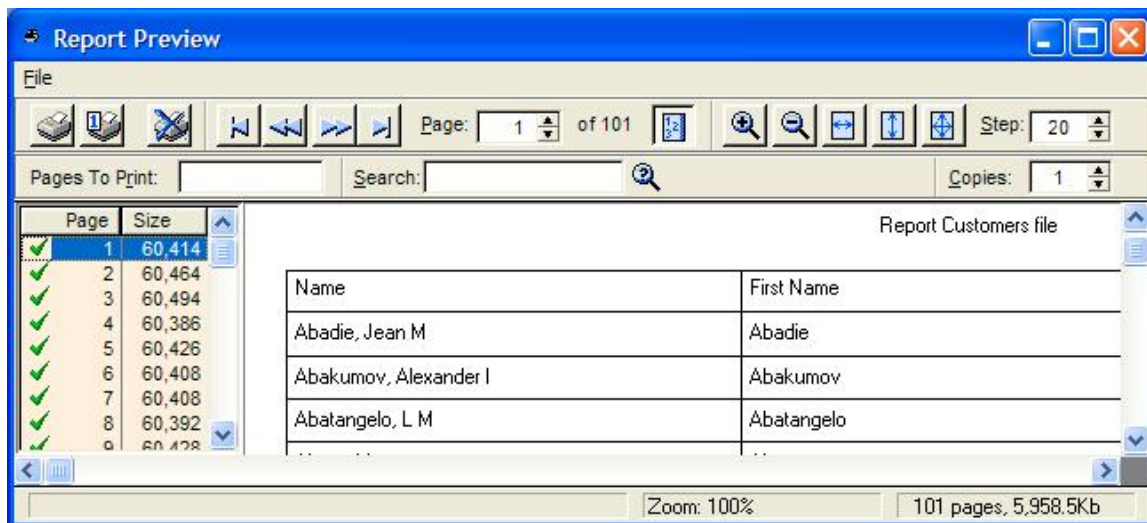
30. Double click on the IcetipsPreviewer procedure to bring up the Procedure Properties window.
31. Click the "Extensions" button.
32. Locate the "IcetipsPreviewer - Icetips Previwer template" class.
33. Select the "ITPClarion6Support - Icetips Previewer - Support for Clarion 6" template.
34. Click the "Select" button.



35. You have now added the Clarion 6 support template to the Previewer.
36. The defaults are all filled out properly
37. If you want to change the icons or tooltip text for the various export types, you can do so now.
38. Click the "Ok" button to back out to the Procedure Properties window
39. Click the "Ok" button to back out to the Applicaton tree window
40. Compile and run.

Done and ready to go

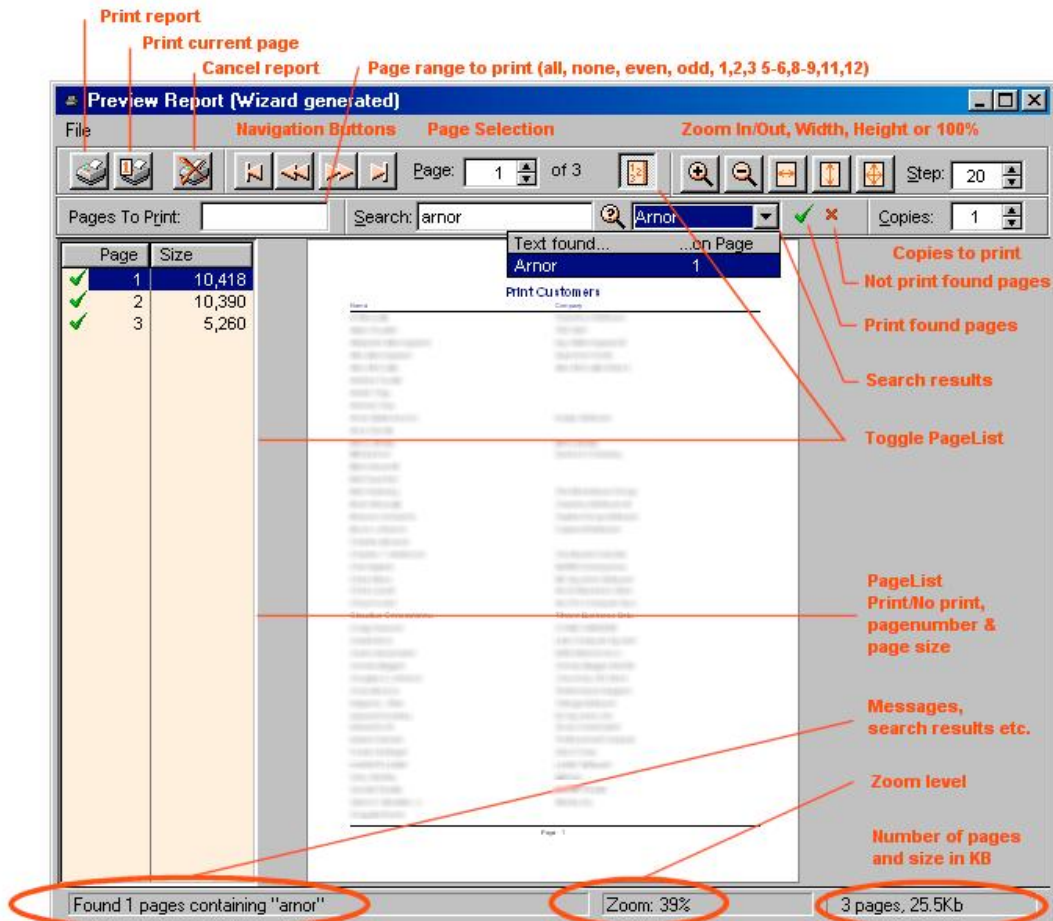
Congratulations! You have now created your first Icetips Previewer procedure!





## The Previewer Window

The Icetips Previewer comes fully loaded with a lot of functionality, much more than the standard Legacy or ABC previewers. The image below shows in one clear image the main functionalities in the Previewer as it is in version 1.000 (released in January 2002). The main interface of the Previewer has not changed much, so we are still using the original image.



The Icetips Previewer compiled in a Clarion 4, Legacy program

## Toolbar

There is a two "row" toolbar at the top. The top row contains navigation, page selection, and zoom levels. The second row contains page selection, searching and copies to print. The pagelist on the left shows all the pages in the report, double click to toggle between print and not print.

## Pages to Print

Pages to print supports words like All, None, Even, Odd as well as individual pages in comma separated lists, like 1,2,3. It also supports ranges, like 1-3 and any kind of combinations:

1,2,3,5-7,9-12,15,18,20-22. If the user gets the ranges wrong, like 12-10, it will automatically "fix" it to be 10-12. Invalid numbers will either be interpreted as the first page or the last page, example 0-5 in a 3 page report will be automatically corrected to 1-3.

## Search

The Search uses a simple Instring() search combined with Upper() so it is case insensitive. On a 1.0GHz machine it takes about 1 second to search through a 40 page report where each page is about 20K. If there are a lot of images on the report, you may consider checking for the total size of the report before you let the user search it. If the search text is found on a page, the page is added to the search page list and then the rest of the page is skipped. When a page is selected from the search results drop down listbox, the previewer automatically jumps to that page. The two buttons next to the search drop down toggle all the pages in the list to be printed, or not to be printed. While the search is running, a progress bar pops up where the dropdown is in the screenshot. The dropdown and the two buttons are only visible if the text is found, and are hidden when the previewer starts.

## Copies

The Copies to Print works in a very simple way as it simply copies the ImageQueue entries x number of times. The way it is set up it will print each copy of the report then the next copy (collated printing). It is a matter of changing a few lines of code to have it print x copies of page 1 then x copies of page 2 etc. if this required.

## Zoom, Step

Zoom is very flexible and by default in the range of 25% to 300%. The zoom step can be changed on the toolbar in steps of 5%. Two buttons allow to zoom in and zoom out, 3 next buttons zoom to page width, page height or 100%. Left click on the page to zoom in, right click to zoom out.

## Toggle / PageList

The Toggle button that is in the top left corner is only visible when the pagelist is hidden. It simply shows and allows the user to toggle if the page should be printed or not. Green checkmark and the page prints, red cross and the page will not print.

*When the previewer starts, the ImageQueue passed to it is moved to a local queue which contains information about size and the toggle field. When the previewer is closing, the pages that are marked not to print are simply not placed into the ImageQueue before it returns.*

*If you have selected to print more than one copy, the pagequeue is simply added x number of times to the ImageQueue before the previewer returns. If you use this option, please remember that the report is printed, then printed again, etc. You don't get x copies of page 1, then x copies of page 2, etc. By modifying the code in the MoveQueueFromLocal routine you can easily do this yourself.*

## Full source provided

When the Icetips Previewer Wizard creates the Previewer procedure, it includes all the source and it's all there for you to use, modify, add to - do whatever you want, once you have created the procedure it is YOUR previewer! The code for Legacy and ABC is very similar, the only difference is a few embeds that use the Init, Kill and TakeWindowEvent methods of the WindowManager object. Other than that, the code is all in control event embeds, window event embeds, data embeds and routines. Most of the actual code is in routines (18 routine embed entries).

Some of you may ask why we didn't write this as OOP? The reason is extremely simple: This way everything is included in one file and when you have created a previewer procedure it is all in the same procedure, nothing external to worry about, it's all there, right at your fingertips to play with!

We have tried to design the code and the window so it's in logical entities. The buttons and controls are grouped together into Groups on the toolbar and usually there is only two or three lines of code in each embed, if it was more than that

Clarion 5.5 Embed tree showing some of the Previewer embeds with code

we moved it into routines that are called instead. There are some exceptions, particularly for alert key event handling on the window and some of the controls on the window (not the toolbar).

## Previewer code and design



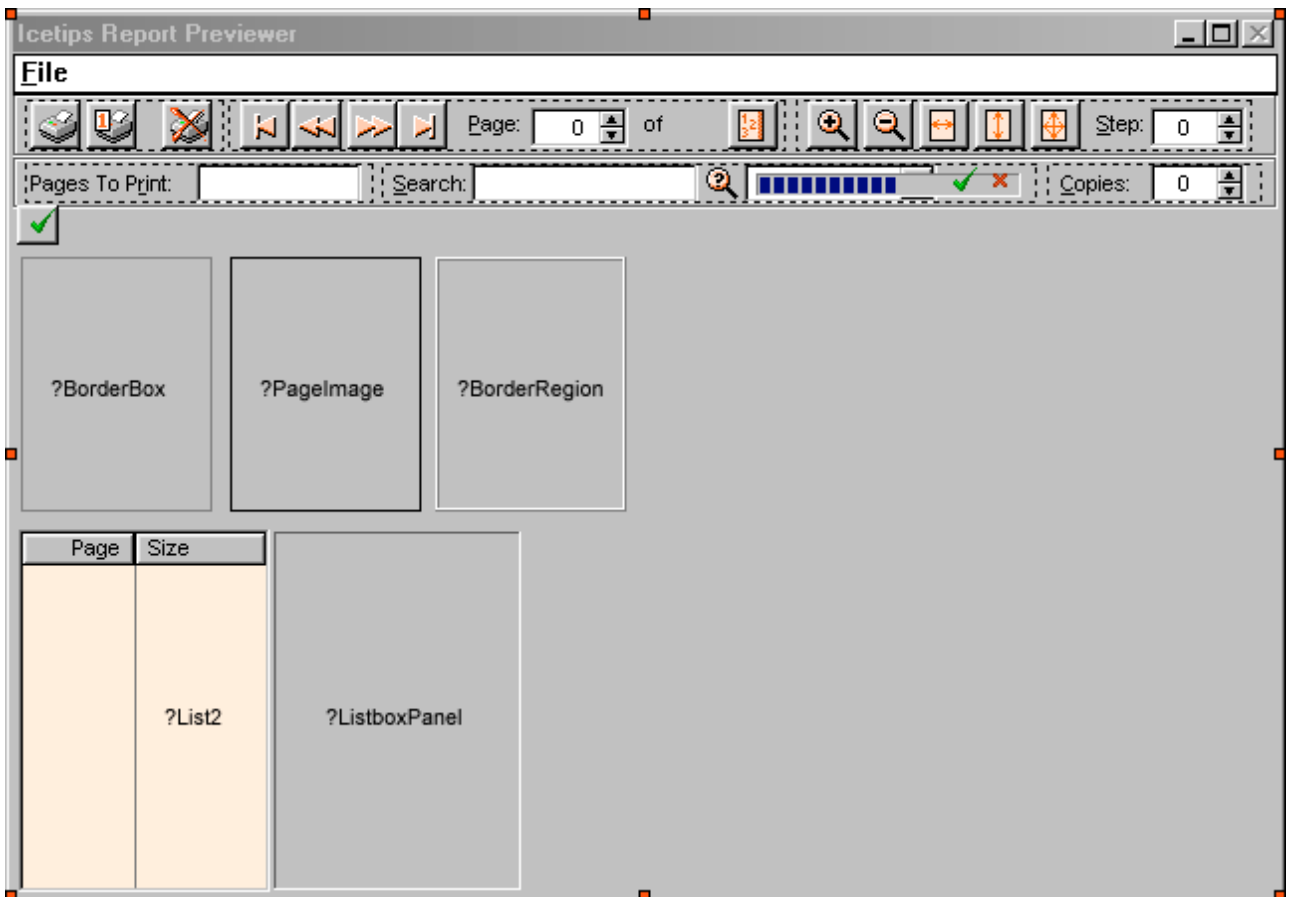
Oops!

Techno stuff follows...

If you don't want to know you don't need to read.

*In-dept discussion follows...*

This section will not discuss the actual code for the previewer, but we will discuss some important things to note about how the previewer works and how it is designed.



*Preview window in Clarion 5.5 Window Formatter showing standard window controls for the Previewer*

## Prototype and Parameters

The Previewer procedure takes several parameters that determine how the previewer works. The prototype is:

```
(*Queue pImageQueue, Short pZoom, Byte pMaximize, String pWindowCaption, Byte pStartPageList),Byte
```

Let's look at each of the parameters passed to the previewer:

| Parameter                        | Function  |
|----------------------------------|---|
| <b>PImageQueue</b>               | This is the queue passed to the previewer which contains the name of the windows metafile image. At the end of the procedure, this queue is free'd and filled again with the pages that should be printed.  |
| <b>PZoom</b>                     | The pZoom parameter specifies what type of zoom should be applied when the previewer starts. Any positive value above 0 is interpreted as an actual percentage zoo, example if you pass the value 50, the zoom will be 50% at startup. Three values have special meaning: -2 (negative two) will zoom to PageHeight, -1 (negative one) will zoom to PageWidth and 0 will zoom to 100% |
| <b>PMaximize</b>                 | If True the previewer window is maximized on startup. If it is not True, the previewer will use whatever setting there is in the ini file like any other window in your application or use the initial size.  |
| <b>PWindowCaption</b>            | Specifies the caption you want to use for the preview window. The default caption of the window is "Icetips Report Previewer" but the default passed to the previewer by the extension template is "Preview Report"   |
| <b>PStartPageList</b>            | Specifies if the pagelist should be visible when the report starts. The default by the extension template is True.  |
| <b>ITP:Response</b> (returned)   | The value of ITP:Response which is the return value of the Previewer is set to True only if the user clicks on either the ?PrintAllButton or the ?PrintOneButton button. This is used by the extension template to determine if the report should be flushed to the printer or not.   |
| <b>&lt; pTargetSelector &gt;</b> | Omittable parameter, used only in Clarion 6. This parameter is used for the Clarion 6.0 export target classes and is required if the Clarion 6 Save Buttons templates are used in the Previewer window.   |



Change  
Prototype

If you want to change the prototype, for example pass some more information to it, there is nothing that prevents you from doing that. The extension template that you apply to your reports to call the previewer both has options for you to type in additional parameters and also to not generate code to call the previewer. If you don't generate the code, it is actually still generated, but it's commented out, so you can see exactly where it is and how it looks like in the embeds around it. If you want to modify the template to support additional parameters as default, you only need to modify the line that creates the %ITPPreviewCall symbol as it is always the same. Look for a line that starts like

```
#SET(%ITPPreviewCall,...
```

and modify it to suit your needs.



**New Parameters**

If you add parameters to the previewer, it would be a good practice to make those parameters omissible by enclosing them in the <> like:

```
(*Queue pImageQueue, Short pZoom, Byte pMaximize, String pWindowCaption, Byte pStartPageList, <BYTE pMyParam>, <STRING pReportName>), Byte
```

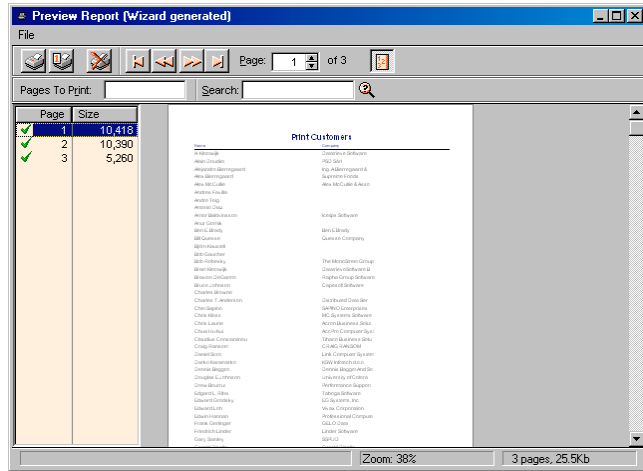
that way you could still use that same previewer for reports that don't pass the additional parameters. Note that the Clarion 6 support extension template adds an optional parameter to pass the ReportTargetSelectorClass from the report to the Previewer to enable the previewer to export the report.

## Changing the Previewer

You can modify the Previewer as much as you like. The source is all there and there is nothing hidden from you. But that said, you need to exercise care when you start modifying and make sure that you don't remove vital parts of the product or it will not work. All the controls on the window itself (not the toolbar) are required except the listbox, ?List2, and the ?ListBoxPanel which are only used for the PageList. The ?GroupBox control provides a visual border around the page. Do not remove it, but you can change its properties if you want. The ?PageImage is also required as it shows the pages. The ?BorderRegion is used to trap the right and left mouseclicks on the page for the zoom to work.

## Removing functionality

If you want to remove functionality from the previewer, the safest way is simply to hide the controls or the entire group of controls if that is what you want. For example if you want to disable the entire navigation group, just hide it and the group is gone. Then you can move the groups around and get the hidden one(s) out of the way for example move the hidden groups over to the right.



*Previewer with the ZoomGroup and SizeGroup hidden*

On the toolbar there are two panel controls that extend horizontally over the entire window. At the very top of the toolbar is a black line that also extends over the entire window. All the groups, except the ones further to the left, have a bevel style of 24576, which shows up as a separator between the groups on the toolbar when the previewer runs. This is set at runtime, in the SetupWindow routine. If you do not want the separators to show up, set the values there to 0 or comment out the lines.



**Group Separators**

In the Data embed a few constants are defined that affect how some parts of the Previewer work. Let's take a brief look at these constants:

! Constants

```
! Minimum and Maximum zoom values in percent
ITP:MinimumZoom EQUATE(25)
```

```
ITP:MaximumZoom EQUATE(300)

! Zoom Constants
ITP:ZoomPageHeight EQUATE(-2)
ITP:ZoomPageWidth EQUATE(-1)

! Events
ITP:EVENT:Sized EQUATE(501h)
```

The first two constants set the minimum and maximum zoom that the previewer will allow. You can change it to whatever you want. The next two are used to equate the two zoom values that can be passed and are not actual zoom percents, that is -1 and -2. There is no need to change those values. The last equate defines a user event that is used to draw the page when the EVENT:Sized is fired. The reason for this extra event is that to get the size of the window clientarea reliably in Clarion 4, we had to create a control on the fly as it did not support PROP:ClientHeight and PROP:ClientWidth as Clarion 5 and Clarion 5.5 do. These two properties are not documented, but are used in the Clarion 5 and Clarion 5.5 report previewers. The problem we ran into was that EVENT:Sized is modal so controls can not be created during the event. We get around that by posting our own event from EVENT:Sized and do the processing there.



*Clarion 4  
compatibility*

## Adding functionality

Adding functionality to the Previewer is only limited by your imagination! There are no limits imposed by the design of the previewer procedure and you can add as much, or as little, to the previewer as you like. Even if you mess everything up, and can't get anything to work any more, you can always create a new previewer and start over.

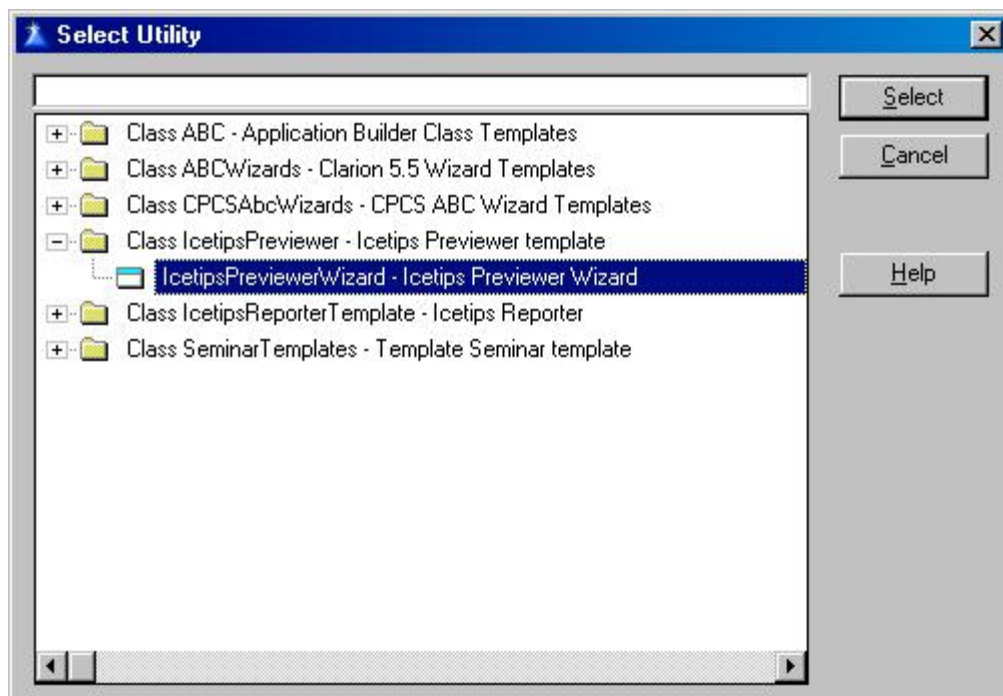
We put as much code into routines as we could to make it easier for developers to make modifications.

## Wizard template

The Wizard template that is provided with the Previewer and creates the Previewer procedure is a very simple one. The only thing it needs to know is the name of the procedure. Note that you cannot create a new procedure and select the Previewer wizard from the wizard list, because it is not tied to any procedure template.

### Run the Wizard

To run the wizard start by selecting "Application|Template Utility" from the main menu in



*Selecting the Icetips Previewer Wizard*

Clarion. Locate the "Class IcetipsPreviewer - Icetips Previewer Wizard" from the list and then select the "IcetipsPreviewerWizard - Icetips Previewer Wizard" from the list.

### Enter Procedure name

Click on the Select button and the Icetips Previewer Wizard pops up and prompts you for a name for the procedure.





*Previewer Wizard - Prompting for Procedure name*

Type in the name of the procedure and click on the Next button.

## Visual theme selection

The Wizard template allows you to select from one of 5 predefined color themes for the previewer. These themes affect the colors of the icons that are used on the buttons in the previewer toolbar. The themes are Orange (default), Yellow, Blue, Green and Gray.



*Selecting a theme file or pick one of the predefined themes*

## Selecting a Theme file

You can also use the Icetips Previewer Editor, discussed later in this document, to create theme files, which allow you to set up the icons, change the window background etc. When

you select the "Select theme file" in the wizard it allows you to select a theme file, which has .itp as a default extension, and use that instead of the predefined themes. Do not forget that whatever you select here, you can change when you have created the previewer, this is simply to make it easier for you to do so.



*Generate Clarion 6 compatible prototype*

## Clarion 6 Options

The Generate Target parameter adds the ReportTargetSelectorClass parameter automatically to the new Previewer procedure. This parameter is needed for support of the Clarion 6 target classes. It can easily be added to the previewer after you create it, but if you are going to be using it, you might as well add it here automatically. You do not need it if you are not using the target classes, but it will not cause any problems if you do have it without using it since it is omissible. This section is only shown if you run the Wizard in Clarion 6.0

## Clarion 6 compatibility

Version 2.0 contains templates that allow the previewer to pass the target selector class from the report to the previewer. This additional parameter makes it possible to export the report currently being viewed in the reporter to PDF, HTML, XML or TXT files. You can always add this parameter too if needed. It is set up as omissible parameter so it does not cause problems with reports and application that do not have the target classes implemented.

## Fully customizable

The Wizard will now create your previewer procedure for you and you can now use it from your reports by applying the Extension template to the report procedures. The Extension template is discussed in the next section.

Remember that when the Wizard has created your Previewer procedure, you have access to everything in it, including a window with over 40 controls on it, almost 1,000 lines of embedded code that make it all work and almost 100 icon files to go with it all! You don't like something? Go change it!

## Extension Templates

The Icetips Previewer contains a few extension templates that are used in various areas of the application, global, on the reports and some are used on the previewer procedure itself to add standard functionality. Version 2.0 also includes three control templates that are used to implement new features/support for Clarion 6 export and PDF-Tools.

The main extension template is the one that is added to reports that use the Previewer. In order to use the previewer on a Report, you must apply that extension template to the Report procedure. The other extension templates are support templates and are not required in order for the previewer to run, but add functionality and options to the previewer itself.

You might ask why we didn't have a global template that automatically cascaded the report extension so that it was automatically added to all your reports. The reason for that is that because of limitations in Clarion we do not do that in any of our products. If you would copy a report procedure from one app to another and both apps had the global template, you would end up with TWO instances of the Previewer template on the procedure that you copied into your app. Example:

```
App1.app
Main
  MyReport
    ITPreviewer
```

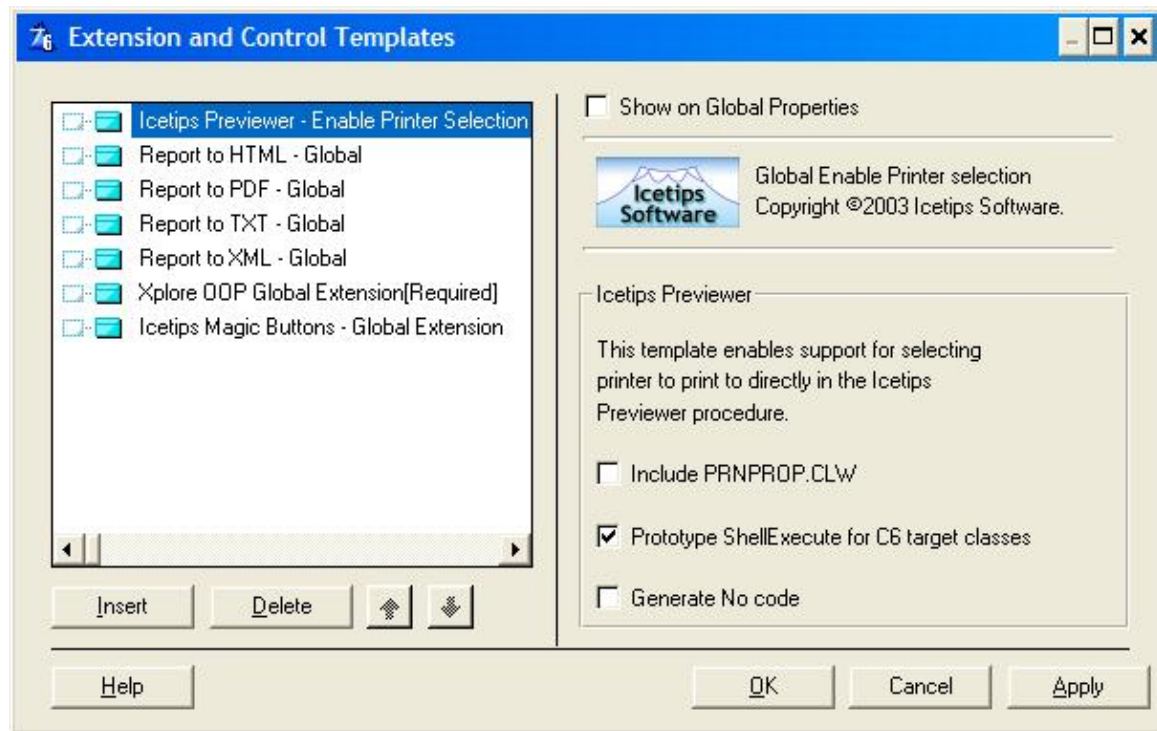
```
App2.app
Main
  MyOtherReport
    ITPreviewer
```

If you copied MyReport from App1.app to App2.app, you will end up with TWO instances of the previewer template on MyReport in App2.app. The problem is that you can not remove the extension from MyReport. The only way to clean it, is to export the procedure to TXA, manually deleting the extra extension template and then import the TXA back in, overwriting the MyReport procedure. We feel that we do not want our customers to find themselves in this situation!

## Global Extension

### Enable Printer Selection

This Global extension template, introduced in version 2.0, has three checkboxes on it. One is



*Icetips Previewer Global Extension Template*

used for the Printer Selection template that you can add to the Previewer and the other is used for Clarion 6 target classes. The last one makes it possible to turn all code generation for this template on or off.

#### Include PRNPROP.CLW

This option is used by the "Enable Printer Selection" extension template that can be applied to the Previewer procedure. It let's the user print a page or report directly from the previewer without leaving it. In order to do so, it must have access to equates that are declared in the PRNPROP.CLW file.

#### Prototype ShellExecute for C6 target classes

This is only available in Clarion 6.0 and allows you to automatically prototype the ShellExecute API so it can be used in the previewer to start the associated program when exporting to a particular target. The API call is prototyped as ITPREV\_ShellExecute which is the name that is used in order to avoid duplicated prototypes. This also means that you can use this ITPREV\_ShellExecute anywhere in your application to run associated programs if you want to. For more information about ShellExecute, refer to Microsoft online MSDN reference at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/reference/functions/shellexecute.asp>

Generate no code

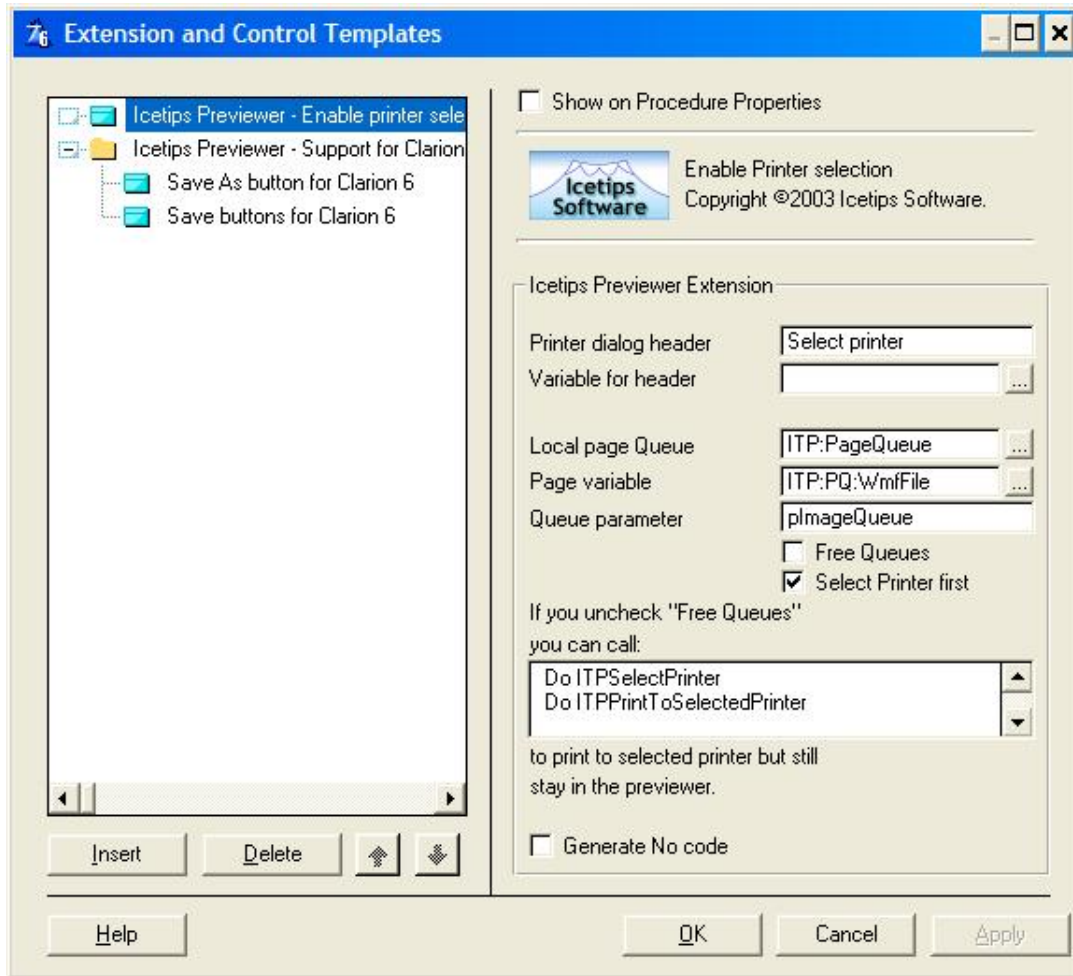
Check this if you want no code generated by the Global Extension template.

## Previewer Extensions

There are two new extension templates that are designed specifically for the Icetips Previewer and do not apply to any other kind of procedures.

### Icetips Previewer - Enable printer selection

This template is used on the Previewer procedure to enable the user to print directly from the previewer.



*Enable printer selection on the Previewer procedure*

#### Printer dialog header

This specifies the header text to use for the PrinterDialog that is called to select the printer.

#### Variable for header

Allows you to use a variable for the Printer dialog header. This is useful for example if you need to support multiple languages and it's easier to store the header text in a variable.

#### Local page Queue

This is the queue used by the previewer to hold the pages. Normally you would simply accept the default which is the ITP:PageQueue

#### Page variable

The variable of the Local Page Queue that contains the actual page filename. Normally you would simply accept the default which is the ITP:PQ:WMFFile variable.

#### Queue parameter

The queue parameter passed to the previewer procedure containing the pages to be printed. Normally you would simply accept the default which is the plmageQueue

#### Free Queues

By checking this the ITPPrintToSelectedPrinter and the ITPPrintCurrentPageToSelectedPrinter routines will free the queues. You would use this to print to the selected printer and close the previewer. If you want to stay in the previewer after you print to the selected printer, uncheck this option.

#### Select Printer first

By checking this, the ITPPrintToSelectedPrinter and ITPPrintCurrentPageToSelectedPrinter routines will call the PrinterDialog before printing. If you uncheck this, it is up to you to let the user select the printer before you call PrintToSelectedPrinter. You can do that by calling ITPSelectPrinter routine.

#### Generate No code

If you check this, no code will be generated by this template.

#### New Local data

This template creates 2 local variables:

```
Loc:OriginalPrinter String(255)
Loc:CurrentPrinter String(255)
```

It also creates a report structure:

```
ITPReport REPORT,AT(0,0),PRE(ITPRPT),THOUS
ITPDetail DETAIL,AT(,,),USE(?ITPDetail)
        IMAGE,AT(0,0),USE(?ITPReportImage)
        END
        END
```

This report structure is used to print the existing pages on to the selected printer.

#### New Routines

This template creates 3 routines:

##### ITPPrintToSelectedPrinter

This routine is used to print to a previously selected printer, or if the "Select Printer first" is checked, it will prompt the user to select a printer before printing.

##### ITPPrintCurrentPageToSelectedPrinter

This routine is used to print the currently visible page to the selected printer or if the "Select Printer first" is checked, it will prompt the user to select a printer before printing.



### ITPSelectPrinter

This routine calls PrinterDialog to select a printer to print to.

## How to implement printer selection

To implement this in your application you need to:

1. Add the Global template to your app, optionally checking the "Include PRNPROP.CLW" option.
2. Add the Procedure extension template
3. Change the code on the ?PrintAllButton button from:

```
ITP:Response = True  
Post(EVENT:CloseWindow)
```

To:

```
Do ITPSelectPrinter  
! Depends on if you checked Select Printer First  
Do ITPPrintToSelectedPrinter  
Post(EVENT:CloseWindow) ! Optional
```

4. Change the code on the ?PrintOneButton button from:

```
Do PrintOne  
ITP:Response = True  
Post(EVENT:CloseWindow)
```

To:

```
Do ITPSelectPrinter  
! Depends on if you checked Select Printer First  
Do ITPPrintCurrentPageToSelectedPrinter  
Post(EVENT:CloseWindow) ! Optional
```

## Icetips Previewer - Support for Clarion 6

This extension template is only available in Clarion 6 and only if one or more of the Clarion 6 target templates have been added to the global section.

This extension sets the previewer procedure up so it can be used to export the report to the various targets available to Clarion 6 users. Currently it supports all the available targets, which are PDF, HTML, XML and TXT.

## How to implement Clarion 6 support and export options

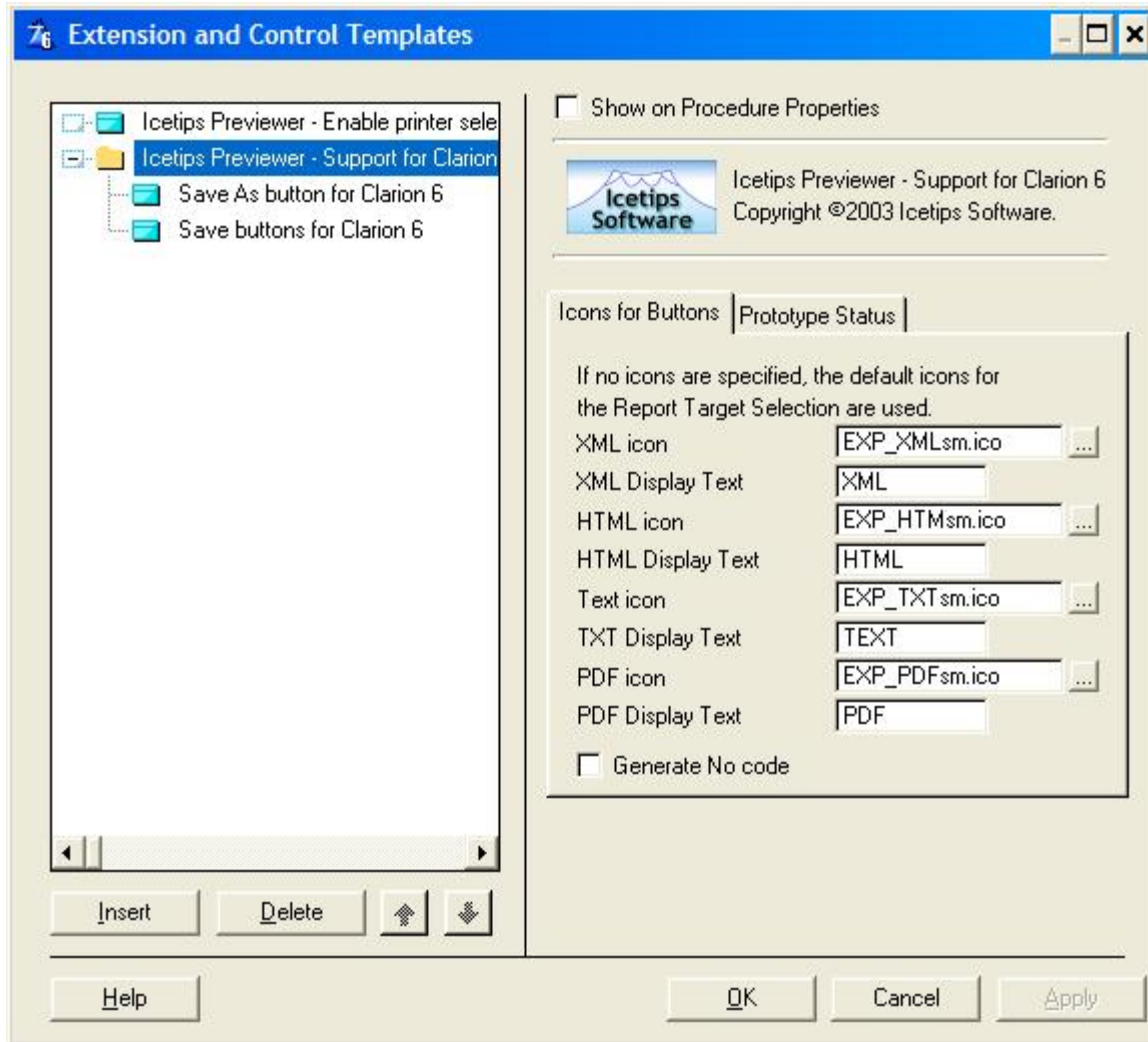
What do you need to do to implement the Target classes in your previewer?

1. Add the SoftVelocity target template(s) to your global Extension area.
2. Add the global template.
3. Add the Support for Clarion 6 support template to the Icetips previewer procedure.
4. Optionally select new icons on the Clarion 6 support template
5. Add either "Save buttons for Clarion 6" or "Save As button for Clarion 6" control templates to the previewer window. You may need to make it wider. If you are using the new wizard template in version 2.0, you do not need to make the window wider.



This will add the export options to your previewer.

The Clarion 6 support template the icons you select to the application project, so you do not need to do that manually. It also adds include files to the module where the Previewer is and creates local data for the Previewer procedure. When you apply the template it checks



*Support for Clarion 6 extension template*

automatically the prototype for the previewer procedure. For the export classes to work you must add:

```
<*ReportTargetSelectorClass pTargetSelector>
```

to the prototype for the previewer. The standard prototype should therefor look like this:

```
(*Queue pImageQueue, Short pZoom, Byte pMaximize, String pWindowCaption, Byte pStartPageList, <*ReportTargetSelectorClass pTargetSelector>),Byte
```

In order to check the prototype, go to the Prototype Status tab and it will inform you if the prototype is correct or if it needs to be corrected. Note that if you have additional parameters in the prototype, the template should still check this correctly. It checks for the

"<\*ReportTargetSelectorClass pTargetSelector>" string in the prototype and if it is not there, it will show an error with information about how to construct the prototype correctly.

You can select different icons to use for the 4 buttons that are created for the 4 different export options in Clarion 6.0. The "Display Text" is used as tooltips on the buttons. They default to PDF, HTML, TEXT and XML.

By checking the Generate No code, no code will be generated by this template.

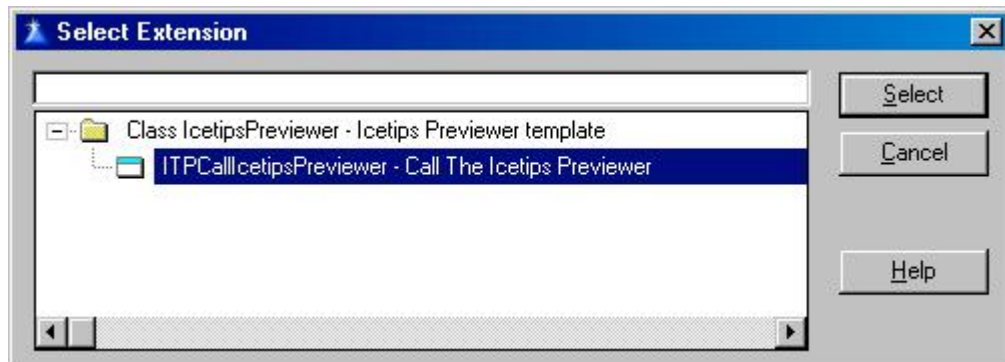
For more information about the buttons, please refer to the section about the Control templates on page 41.

## Report Extension

In order for you to use your new previewer procedure in your application, you need to apply an extension template to your report procedures. You can only apply the extension template to report procedures.

The Report Extension is applied to the report that should use the Icetips Previewer.

## Apply the Extension



*Selecting the Call The Icetips Previewer Extension template*

Select Extension template

Select the "Call The Icetips Previewer" from your list of Extension templates. This applies the Extension to the report and will generate all the code for you that is needed to call your previewer.



*Using multiple previewers*

*In-dept discussion follows...*

You can add as many instances of this extension template as you want, each calling a different previewer, or with different parameters. Each condition is validated separately, in an If statement. The logic of the generated code is very simple:

```

If FirstCondition
  Previewer1
End
If SecondCondition
  Previewer2
End
  
```

And so forth. This does mean that you can have FirstCondition and SecondCondition both be true, which will results in both previewers running and the second one will probably not have any pages to display because after the first one finishes, it prints or discards the pages, so there's nothing left for the second one!



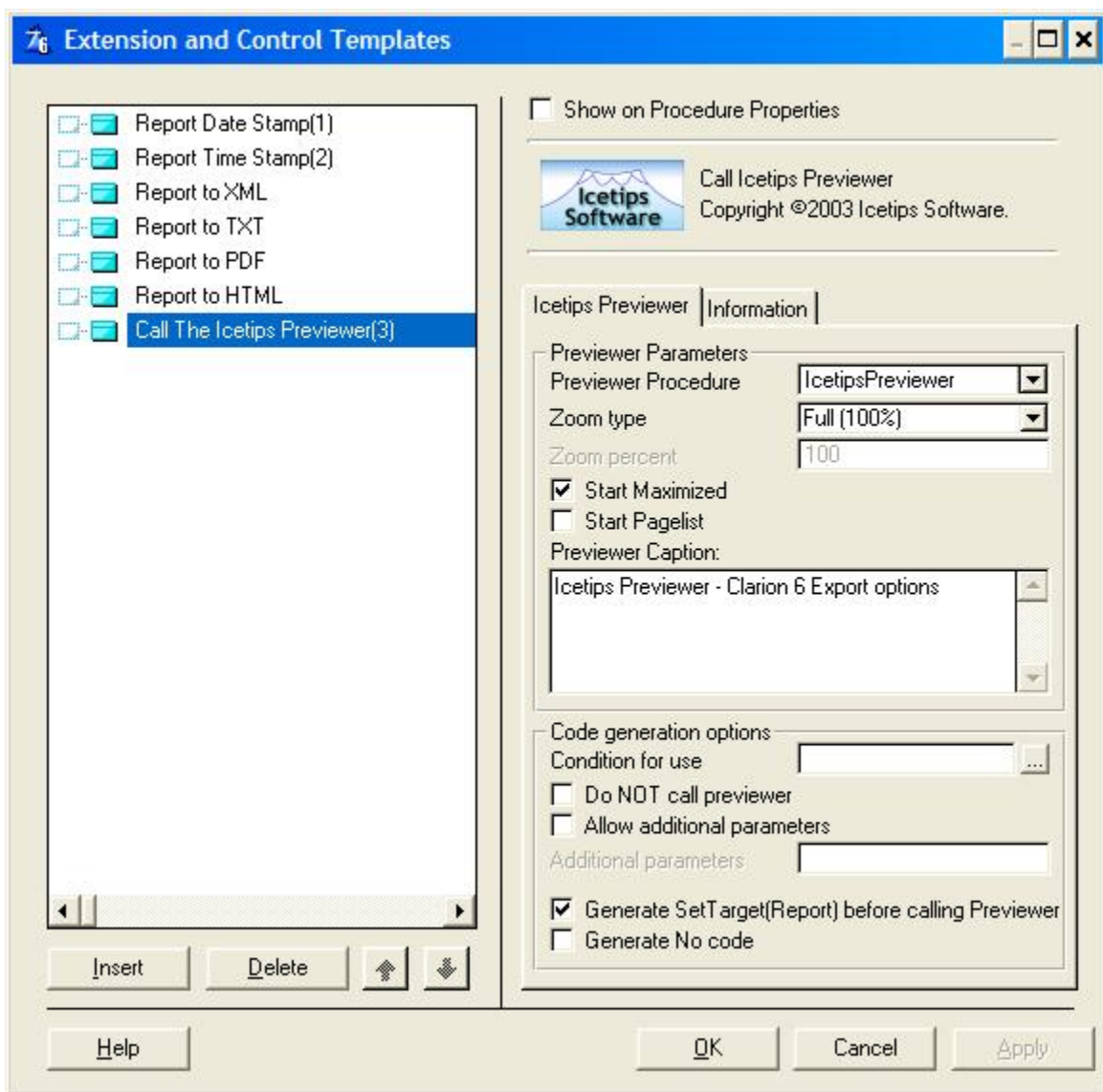
*Using the standard previewer as an option*

The Demo program supplied with the Icetips Previewer shows a simple way to deal with this as it sets one variable and then this one variable is used in the conditions. It shows how to call a selection window where you can select the previewer you want to use and then use a variable to conditionally run the previewer of choice. Note that Clarion's or

CPCS' standard previewer is never called if you apply the extension templates. If you want to use Clarion's or CPCS' standard previewer as an alternative, you need to handcode it in the appropriate embeds. The Demo program shows how to do that in a simple way.

*Note that in Clarion 4 ABC, the code generated is actually Legacy code. After spending days on trying to find a solution to an obscure problem when importing TXA for Clarion 4 ABC procedure with all the embeds, we finally gave up and used the Clarion 4 Legacy TXA instead, which works perfectly in Clarion 4 ABC! Clarion 5 and Clarion 5.5 had no problems with the same TXA code.*

When you have applied the extension template, you only need to select the previewer procedure from the dropdown list, where the template prompts you for "Previewer Procedure".



*Icetips Previewer Report Extension Template*

## Icetips Previewer Tab

### Zoom type

The Zoom type set's the initial zoom that is used when the previewer is called. This can be "Page Height", "Page Width", "Full (100%)" or "Value". If you select the Value from the drop down, the Zoom percent field is enabled and you can type in whatever percent value you want the previewer to start in.

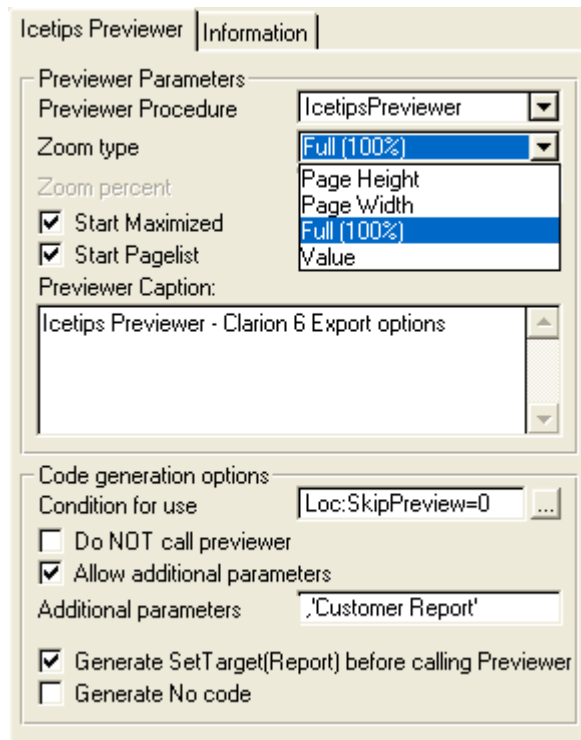
The two checkboxes, Start Maximized and Start Pagelist affect how the previewer starts.

### Start Maximized

Start Maximized will maximize the window when it runs. If this is not checked, the previewer will start in either the designed size (which should fit in a 640x480 window, though it may be a bit tight...) or at whatever size was saved for it, if the window size/position is being saved to an ini file. Start Maximized is checked by default.

### Start Pagelist

The Start Pagelist specifies if the PageList should be started when the previewer window opens or not. Start Pagelist is checked by default.



*Zoom type drop down on Icetips Previewer Extension Template window*

### Previewer Caption

The Previewer Caption sets up the text you want to use as caption for the previewer.

### Code generation options

The group that is called Code generation options sets up the condition for using the previewer. Note that if you set up a condition and it is false and there is no other instance of the previewer, there will be no preview at all!

### Condition for use

The code that you put in to the Condition for use field is used directly in an If statement and if the condition evaluates to True the previewer runs.

The Demo program, supplied with the Previewer, has code that shows how to select a previewer and how to use the standard previewer. Condition for use is empty by default.

### Do NOT call previewer

The Do NOT call previewer can come in handy if you want to hand code the calls to the previewer. What it does is comment the generated code, rather than

not generate it. That way you can see how the generated code looks like when you review the code and edit it in the Embeditor in Clarion. Do NOT call previewer is unchecked by default.

### Allow additional parameters

Allow additional parameters allows you to add parameters to your call to the previewer. This comes in handy if you have added parameters to the preview procedure and need to add your parameters here. Note that these parameters are in addition to the standard, default parameters used by the template. Allow additional parameters is unchecked by default.

Additional parameters

As an example, let's say you wanted to pass an additional parameter to the previewer procedure with a descriptive name of the procedure. You simply check the Allow additional parameters, and type the parameter into the Additional parameters entry field:

```
'Print People information'
```

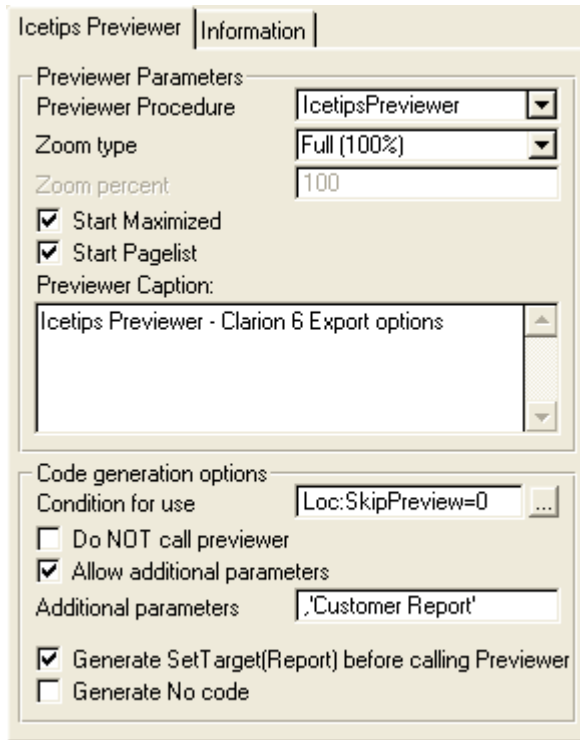
You need a comma as the first character to separate it from the parameter list, but if you forget, the template will put it there for you. This will now generate the following call to the previewer:

```
If ITPreviewer(SELF.Preview.ImageQueue, 100, 1, |
    'Report Preview', 1,'Print People information')
```

Instead of the standard:

```
If ITPreviewer(SELF.Preview.ImageQueue, 100, 1, 'Report Preview', 1)
```

In the previewer procedure, you would add this as ,<STRING pProcedure> or something similar at the end of the standard parameter list, don't forget to update both the Parameter and



Passing Additional Parameters to your Previewer

Prototype for the previewer! Good practice is also to copy the prototype and put it as a comment in one of the first embed points in the procedure, that way you can see it when you are in the embed tree. The Additional parameters field is empty by default and disabled.

Generate No code

By checking this, no code will be generated by this template.

Information tab

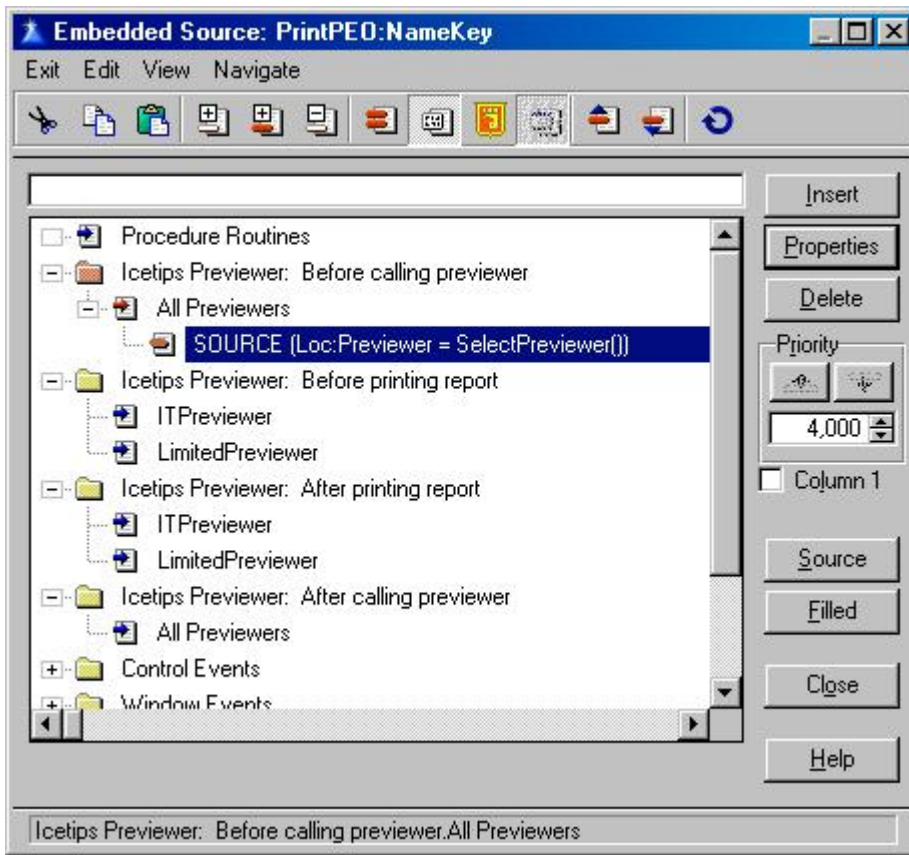
The Information tab only has some generic information about the conditional calls to the previewer procedure. It simply stresses the need to make sure that at least one condition evaluates to true when you conditionally call the previewer or nothing will preview or print.

Previewer Extension Embed points

When you apply the Extension template to your report, it automatically inserts 4 embeds that you can use to do anything you like before and after calling the previewer. Note that CPCS report templates will only have 2 embeds, not 4, due to the different design of CPCS reports.



Two of these embeds are previewer dependent, two are not. Please refer to the image to the left. As you can see there are 2 entries for Icetips Previewer:



Before calling previewer, which has only one instance, the same as Icetips Previewer: After calling previewer.

Then there are two entries for Icetips Previewer: Before printing report and Icetips Previewer: After printing report. Both of these have two instances, one for each instance of the Extension template. The

Embed tree in Clarion 5.5 ABC showing embeds for two instances of Icetips Previewer

first embed, the Before calling previewer, is inserted before any calls to a previewer and this is the perfect place to set conditional variables etc and do anything that needs to be done before the previewers are called. In the demo program, in the PrintPEO:NameKey procedure, this is what you will find in the Icetips Previewer ABC: Before calling previewer embed:



How to conditionally use the standard previewer

```

Loc:Previewer = SelectPreviewer()
If Loc:Previewer = 0                                ! Selection Cancelled
    SELF.Report {PROP:FlushPreview} = False
    Free(SELF.Preview.ImageQueue)
End
If Loc:Previewer = 3                                ! Use the standard ABC Previewer
    PARENT.AskPreview
End
    
```

This is the only code needed to call the selection window and either cancel the report or use one of three previewers available! Note that when you use the Icetips Previewer Extension template, the Clarion or CPCS standard previewer is never called unless you call it manually. If you want to use it you must call it in code. All the code you need is Parent.AskPreview - see the code above. This applies to ABC. In Legacy there is a little bit more code needed. Please read about that in the Compatibility issues section later in this document.

The two embeds that have one instance for each previewer are handy to execute code that is previewer dependent before and/or after the previewer procedure is called. Again note that

the way the embeds show up may differ between versions of Clarion and it is possible that last minute minor changes were made to the embed structure after this document was finished. There may be ever so slight differences between what you see in Clarion and what you see in this document. We try to keep these changes to a minimum and get new screenshots into the documentation as quickly as possible.

## Note about version 1.001

We have rewritten the extension template for Clarion 5 Legacy only in version 1.001, released on February 6, 2002. Some serious problems were discovered in the Clarion 5 Legacy that caused code in Before Print Preview to be generated after some of the code was generated to call the Icetips Previewer. Fixing this for Clarion 5, broke the Extension template in both Clarion 4 and Clarion 5.5 Legacy, so we had no choice but to distribute two extension template files that are renamed appropriately by the installation program after they are installed. This means that if you want to use the Icetips Previewer in Clarion 5 and some other Clarion version, you MUST use the installation program to install it. The Clarion 5 extension template will not work in Clarion 4 or Clarion 5.5 and the Clarion 4/Clarion 5.5 extension template will not work properly in Clarion 5.

## Note about version 2.000

Version 2.000, released in March, 2004 contains a number of new features. Please refer to the "what is new in 2.0" chapter on page 51 and also to the "Previewer Extensions" on page 30 and Global Extension on page 28.



# Control Templates

Version 2.0 of the Icetips Previewer contains three control templates that are used to add support for the Clarion 6 export targets and PDF-Tools.

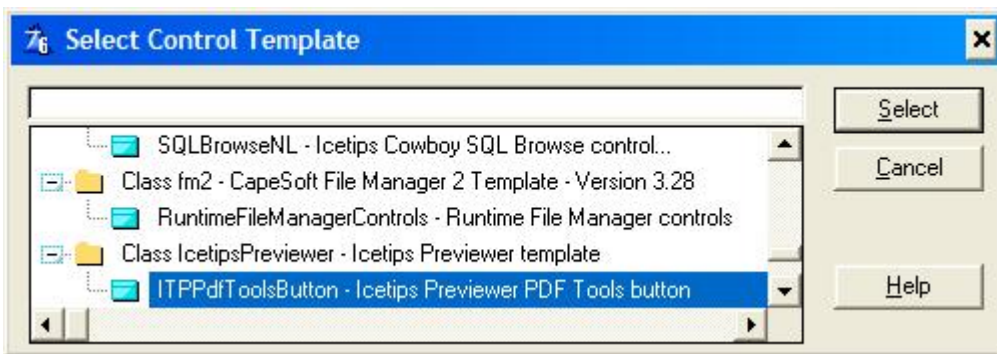
There are two different types of control templates that you can use in Clarion 6 to exort the report to the various target types available. The Save Buttons populate 4 buttons on the window, but the Save As button populates only one button. This button then calls the standard target selector window and exports the report to the selected target.

## Icetips Previewer PDF Tools button

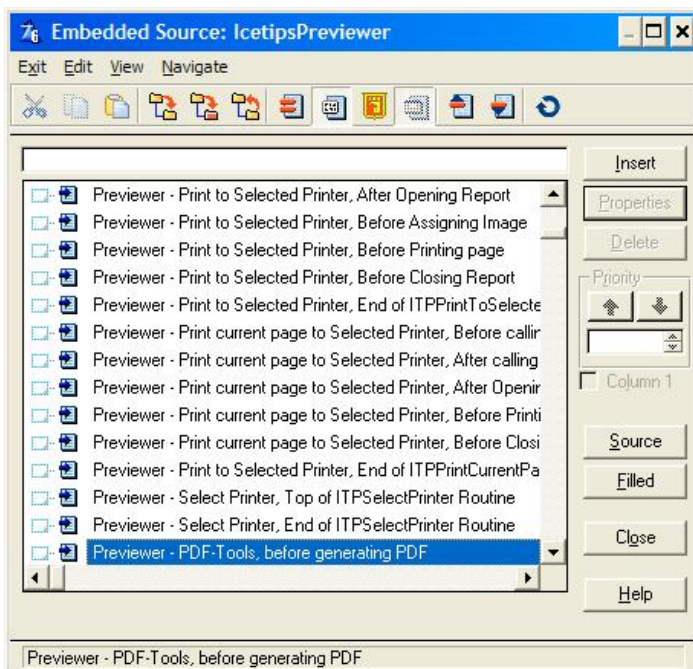


This control template can be used with DocuTrack's PDF-Tools to generate PDF documents from the report that is being previewed. You can visit Docu-Track at their website at <http://www.docu-track.com/index.php>

### Adding the control template



Add the PDF Tools button control template



Open the Previewer window procedure and open the window formatter by clicking on the "Window" button. Then select "Populate | Control template" from the window formatter menu and select "ITPPdfToolsButton - Icetips Previewer PDF Tools button" from the list of control templates.

### New embeds

When you add the control template one additional embed point is created, called "Previewer - PDF-Tools, before

generating PDF. This embed is inside a routine that the template creates, ExportPDFTools25

## Generated code

The control template causes an additional routine to be created in the Previewer Procedure. It also adds an instance of the PDFXToolsReportClass to the Previewer procedure as well as a call to the new routine, ExportPDFTools25. Following is the standard code that this control template generates.

Local data:

```
ITPDFXTR3          PDFXToolsReportClass
```

Accepted event on the ?PDFToolsButton control:

```
Do ExportPDFTools25
```

The template also adds the following routine.

```
ExportPDFTools25          ROUTINE
  Data
  TempQueue   Queue
  PageName    String(128)
              End
  R Long
  I Long
  ShowPDFProgress Byte(1)

  Code
  Free(TempQueue)
  If Not Records(ITP:PageQueue)
    Exit
  End

  Loop I = 1 To Records(ITP:PageQueue)
    Get(ITP:PageQueue,I)
    If ITP:PageQueue.ITP:PQ:PrintPage
      TempQueue.PageName = ITP:PageQueue.ITP:PQ:WmfFile
      Add(TempQueue)
    End
  End
  If Records(TempQueue)
    ITPDFXTR3.Init()
  ! Start of "Previewer - PDF-Tools, before generating PDF"
  ! [Priority 4000]
  ! NEW EMBED
  ! End of "Previewer - PDF-Tools, before generating PDF"
  ! Example method calls:
  !ITPDFXTR3.SetAuthor ('Author')
  !ITPDFXTR3.SetTitle ('Title')
  !ITPDFXTR3.SetSubject ('Subject')
  !ITPDFXTR3.SetCreator ('Creator')
  !ITPDFXTR3.SetKeywords ('Keywords')
  R = ITPDFXTR3.Generate(TempQueue,ShowPDFProgress)
  Free(TempQueue)
  End
```

The embed allows you to set any of the ITPDFXTR3 properties or do whatever needs to be done before the PDF file is generated.

## Save buttons for Clarion 6

This template adds 4 buttons to the previewer window, where you select to drop them. These buttons are dynamically assigned the various output targets at runtime. You should not change the icons or tooltips on the buttons in the window formatter, but change them in the extension template, see the chapter on Clarion 6 support starting on page 32 for more information. The four buttons are grouped together in a group structure so they can be moved together. Unless you only have some of the target templates, we do not recommend that you remove any of the buttons. They are hidden at runtime if they are not available.



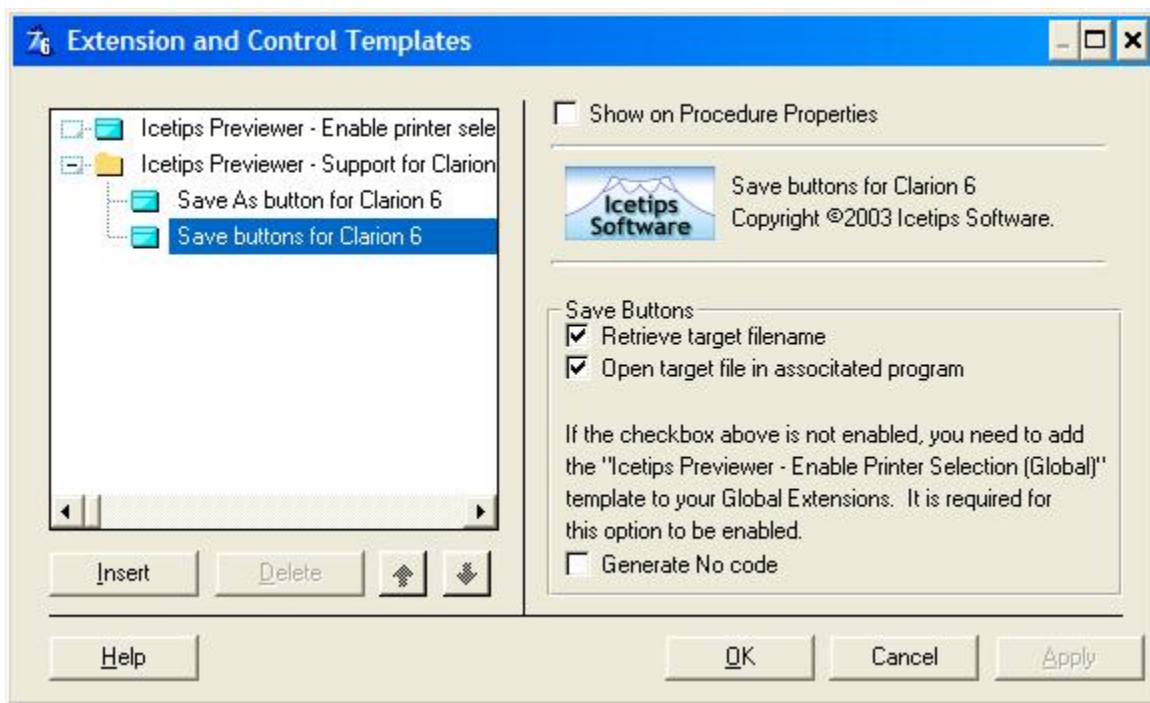
Clarion 6 Save As Buttons



Runtime buttons

The buttons don't look very interesting while they are in the Window formatter, but at runtime they come to life and show up nicely with the appropriate icons that you selected for them.

This template adds two routines to the previewer, EnableC6Buttons and SaveToSelectedC6Format. The first one has no embed points, but the second one has several embed points. When the buttons are pressed the Loc:C6SelectedFormat local variable is set to the appropriate value, 1 to 4 and then the SaveToSelectedC6Format routine is called.



Save Buttons extension template

EnableC6Buttons does all the ground work to find out what targets are available, what icons to use and what tooltips to use. This routine is called from the OpenWindow event handler.

The SaveToSelectedC6Format routine does all the export work for the previewer and then executes the associated program using ShellExecute, if that option was selected. If you want

to put a condition on the ShellExecute, then you can do that in the two embeds before and after ShellExecute.

Retrieve target filename

When this is checked, the filename of the target is retrieved into ITP:TargetFileName variable. This variable is then available to use with ShellExecute if the Open target file in associated program is checked too, or for your own use if you need it.

Open target file in associated program

If this is checked the previewer will attempt to launch whatever program is associated with the file extension of the target file. For example if you export to PDF, it will attempt to load the exported file in Adobe Acrobat. The generated code does not perform any error checking, but you can check the return value of ShellExecute in the "C6 Save buttons: After ShellExecute" embed. The return value is stored in the ITP\_RetHandle variable which is declared in the routine and is therefore only available inside the SaveToSelectedC6Format routine.

Generate No code

By checking this, no code will be generated by this template.

## Generated code

This template generates two routines and a few more lines of code.

OpenWindow Event Handling:

```
Do EnableC6Buttons
```

Control Event Handling, ?Clarion6Button1 - ?Clarion6Button4, Accepted

```
Loc:C6SelectedFormat = 1
Do SaveToSelectedC6Format
```

Procedure Routines:

EnableC6Buttons has no embed points. Several lines are generated based on what targets are available. The following is generated where all the export targets are available.

```
EnableC6Buttons          ROUTINE
  Data
  I                      Long
  NumOpts                Byte
  ReturnValue             Long
  Code
  C6Buttons[1] = ?Clarion6Button1
  C6Buttons[2] = ?Clarion6Button2
  C6Buttons[3] = ?Clarion6Button3
  C6Buttons[4] = ?Clarion6Button4

  If NOT pTargetSelector &=NULL
    NumOpts = pTargetSelector.Items()
    Loop I = 1 To NumOpts
      Clear(GQ)
      GQ.GName = pTargetSelector.GetReportGeneratorName(I)
      Add(GQ)
    End
    Loop I = 1 TO Records(GQ)
      Get(GQ,I)
      C6Buttons[I] {Prop:Hide} = False
      C6Buttons[I] {Prop:Disable} = False
```

```

ITPReportTarget &= pTargetSelector.GetReportGenerator(GQ.GName)
C6Buttons[I] {PROP:ICON} = ITPReportTarget.DisplayIcon()
C6Buttons[I] {PROP:TIP} = ITPReportTarget.DisplayName()
Case Upper(ITPReportTarget.DisplayName())
Of Upper('HTML')
  C6Buttons[I] {PROP:ICON} = 'EXP_HTMsm.ico'
Of Upper('TEXT')
  C6Buttons[I] {PROP:ICON} = 'EXP_TXTsm.ico'
Of Upper('PDF')
  C6Buttons[I] {PROP:ICON} = 'EXP_PDFsm.ico'
Of Upper('XML')
  C6Buttons[I] {PROP:ICON} = 'EXP_XMLsm.ico'
End
End
End

```

The SaveToSelectedC6Format routine is the main routine and does all the target exporting. It has several embed points that can be used to customize the routine to your needs.

```

SaveToSelectedC6Format          ROUTINE
  Data
  I Long
  ReturnValue          Long
  ITP_Handle          UNSIGNED
  ITP_Op              CSTRING(20)
  ITP_Path            CSTRING(256)
  ITP_Param           CSTRING(256)
  ITP_Show            SHORT
  ITP_RetHandle       USHORT

  Code
  If Not Records(GQ)
    Exit
  End
! Start of "C6 Save buttons: Beginning of code"
! [Priority 4000]

! End of "C6 Save buttons: Beginning of code"
Get(GQ,Loc:C6SelectedFormat)
If NOT pTargetSelector &=NULL
  ITPOutputFileQueue &= NEW OutputFileQueue
  ITPPrevQueue       &= NEW PreviewQueue
  ITPWmfParser       &= NEW WMFDocumentParser

! Start of "C6 Save buttons: Before adding to queue"
! [Priority 5000]

! End of "C6 Save buttons: Before adding to queue"
Loop I = 1 To Records(ITP:PageQueue)
  Get(ITP:PageQueue,I)
  If ITP:PageQueue.ITP:PQ:PrintPage
    ITPPrevQueue.FileName = ITP:PageQueue.ITP:PQ:WmfFile
    Add(ITPPrevQueue)
  End
End

! Start of "C6 Save buttons: Before Export"
! [Priority 5000]

! End of "C6 Save buttons: Before Export"
  If Records(ITPPrevQueue)

```

```

    If NOT pTargetSelector &=NULL
        ITPReportTarget &= pTargetSelector.GetReportGenerator(GQ.GName)
        If ITPReportTarget.SupportResultQueue() = True
            ITPReportTarget.SetResultQueue(ITPOutputFileQueue)
        End
! Start of "C6 Save buttons: Before Save Dialog"
! [Priority 5000]

! End of "C6 Save buttons: Before Save Dialog"
    If ITPReportTarget.AskProperties(False) = Level:Benign
        ITPWmfParser.Init(ITPPrevQueue, ITPReportTarget)
! Start of "C6 Save buttons: Before Saving"
! [Priority 5000]

! End of "C6 Save buttons: Before Saving"
        ReturnValue = ITPWmfParser.GenerateReport()
! Start of "C6 Save buttons: After Saving"
! [Priority 5000]

! End of "C6 Save buttons: After Saving"
        If ReturnValue = Level:Benign
            Get(ITPOutputFileQueue,1)
            ITP:TargetFileName = ITPOutputFileQueue.FileName
            If ITPReportTarget.SupportResultQueue() = True
                End
                ITP_Op      = 'open'
                ITP_param   = ''
                ITP_path    = Clip(Path())
                ITP_show    = 1
! Start of "C6 Save buttons: Before ShellExecute"
! [Priority 5000]

! End of "C6 Save buttons: Before ShellExecute"
                ITP_RetHandle = ITPREV_ShellExecute(0{Prop:Handle}, |
                                                                    ITP_Op, |
                                                                    ITP:TargetFileName, |
                                                                    ITP_Param, |
                                                                    ITP_Path,
                                                                    ITP_Show)
! Start of "C6 Save buttons: After ShellExecute"
! [Priority 5000]

! End of "C6 Save buttons: After ShellExecute"
            End
        End
    End
! Start of "C6 Save buttons: After Export"
! [Priority 5000]

! End of "C6 Save buttons: After Export"

    Dispose(ITPWmfParser)                ! Dispose parser

    Free(ITPOutputFileQueue)             ! Dispose output queue
    Dispose(ITPOutputFileQueue)

    Free(ITPPrevQueue)                   ! Dispose preview queue
    Dispose(ITPPrevQueue)
End
! Start of "C6 Save buttons: End of code"
! [Priority 5000]

```



! End of "C6 Save buttons: End of code"

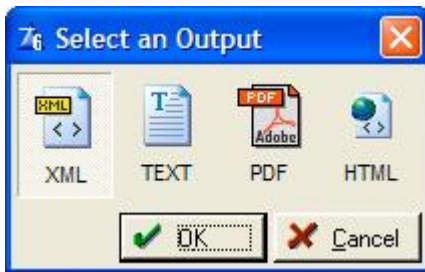
## Save As button for Clarion 6



Save As button

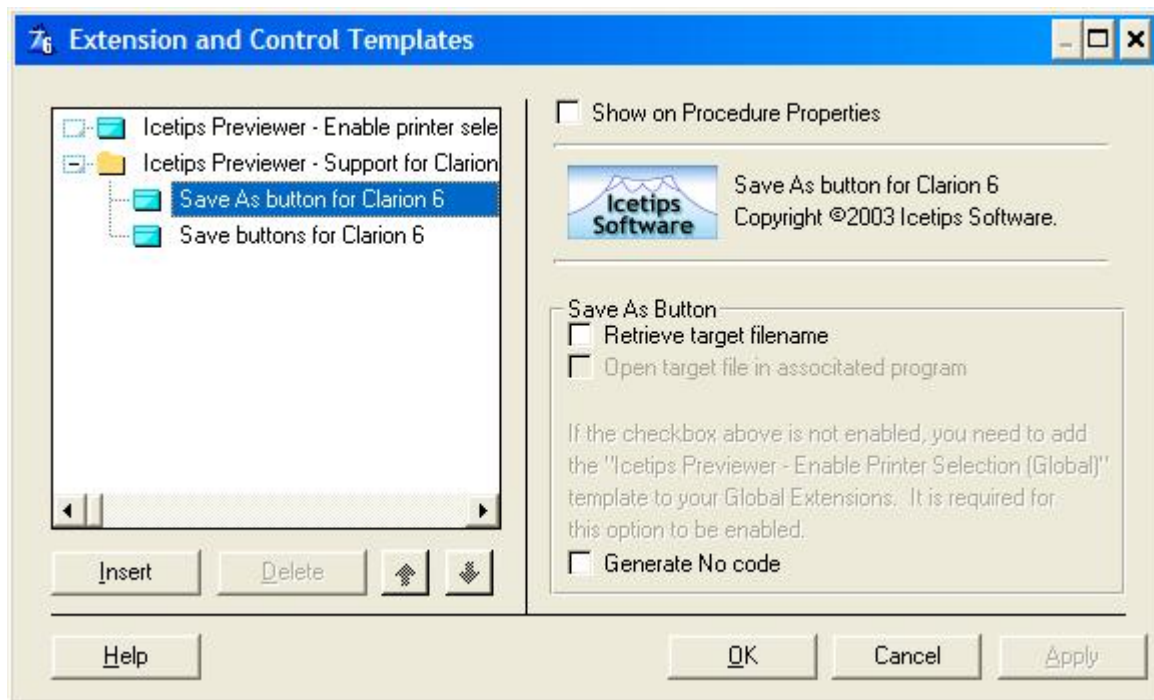
This template adds one button to the previewer window, where you select to drop it. At runtime this button calls the Target selector window where you can select what format to save to.

When this button is pressed it will bring up the Target selection window, see screenshot below. This window is completely outside of the Icetips Previewer and can not be modified by the previewer. If you need to modify this window, we suggest you go through the documentation on the Target classes in Clarion 6.



Clarion 6 target selection window

You can change any properties you want to on this button.



Save As extension template



Retrieve target filename

When this is checked, the filename of the target is retrieved into ITP:TargetFileName variable. This variable is then available to use with ShellExecute if the Open target file in associated program is checked too, or for your own use if you need it.

Open target file in associated program

If this is checked the previewer will attempt to launch whatever program is associated with the file extension of the target file. For example if you export to PDF, it will attempt to load the exported file in Adobe Acrobat. The generated code does not perform any error checking, but you can check the return value of ShellExecute in the "C6 Save As button: After ShellExecute" embed. The return value is stored in the ITP\_RetHandle variable which is declared in the routine and is therefor only available inside the SaveAsClarion6 routine.

Generate No code

By checking this, no code will be generated by this template.

## Generated code

This template generates one routine and a few more lines of code.

Control Event Handling, ?C6SaveAsButton, Accepted

Do SaveAsClarion6

Procedure Routines:

The SaveAsClarion6 routine is the main routine and does all the target exporting. It has several embed points that can be used to customize the routine to your needs.

```
SaveAsClarion6          ROUTINE
  Data
  ReturnValue           Long
  I                     Long
  ITP_Handle            UNSIGNED
  ITP_Op                CSTRING(20)
  ITP_Path              CSTRING(256)
  ITP_Param              CSTRING(256)
  ITP_Show              SHORT
  ITP_RetHandle         USHORT

  Code

  ITPOutputFileQueue  &= NEW OutputFileQueue
  ITPPrevQueue         &= NEW PreviewQueue
  ITPWmfParser         &= NEW WMFDocumentParser
! Start of "C6 Save As button: Beginning of code"
! [Priority 5000]

! End of "C6 Save As button: Beginning of code"
Loop I = 1 To Records(ITP:PageQueue)
  Get(ITP:PageQueue,I)
  If ITP:PageQueue.ITP:PQ:PrintPage
    ITPPrevQueue.FileName = ITP:PageQueue.ITP:PQ:WmfFile
    Add(ITPrevQueue)
  End
End
! Start of "C6 Save As button: Before Export"
! [Priority 5000]
```

```

! End of "C6 Save As button: Before Export"
  If Records(ITPPrevQueue)
    If NOT pTargetSelector &=NULL
! Start of "C6 Save As button: Before Selecting Target"
! [Priority 5000]

! End of "C6 Save As button: Before Selecting Target"
  If pTargetSelector.Ask(False,True)
! Start of "C6 Save As button: After Selecting Target"
! [Priority 5000]

! End of "C6 Save As button: After Selecting Target"
  ITPReportTarget &= pTargetSelector.GetSelected()
  If ITPReportTarget.SupportResultQueue() = True
    ITPReportTarget.SetResultQueue(ITPOutputFileQueue)
  End
  If ITPReportTarget.AskProperties(True) = Level:Benign
    ITPWmfParser.Init(ITPPrevQueue, ITPReportTarget, GlobalErrors)
! Start of "C6 Save As button: Before Saving"
! [Priority 5000]

! End of "C6 Save As button: Before Saving"
  ReturnValue = ITPWmfParser.GenerateReport()
! Start of "C6 Save As button: After Saving"
! [Priority 5000]

! End of "C6 Save As button: After Saving"
  If ReturnValue = Level:Benign
    Get(ITPOutputFileQueue,1)
    ITP:TargetFileName = ITPOutputFileQueue.FileName
    If ITPReportTarget.SupportResultQueue() = True
      End
      ITP_Op    = 'open'
      ITP_param = ''
      ITP_path  = Clip(Path())
      ITP_show  = 1
! Start of "C6 Save As button: Before ShellExecute"
! [Priority 5000]

! End of "C6 Save As button: Before ShellExecute"
  ITP_RetHandle = ITPREV_ShellExecute(0{Prop:Handle}, |
                                          ITP_Op, |
                                          ITP:TargetFileName, |
                                          ITP_Param, |
                                          ITP_Path,
                                          ITP_Show)
! Start of "C6 Save As button: After ShellExecute"
! [Priority 5000]

! End of "C6 Save As button: After ShellExecute"
  End
  End
  End
  End
  End

Dispose(ITPWmfParser)                                ! Dispose parser

Free(ITPOutputFileQueue)                             ! Dispose output queue
Dispose(ITPOutputFileQueue)

```

```
Free(ITPrevQueue)                                ! Dispose preview queue
Dispose(ITPrevQueue)
! Start of "C6 Save As button: End of code"
! [Priority 5000]

! End of "C6 Save As button: End of code"
```

## What is new in version 2.0

### New templates

Version 2.0 introduces some new extension and control templates that add to the functionality of the Icetips Previewer. A global and procedure extension templates add options to select a printer to print to in the previewer and either return to the report or stay in the previewer.

Clarion 6 added options to export the report to various targets, PDF, HTML, Text and XML. We have written extension and control templates to take advantage of those new options in Clarion 6. A single extension template is added to the previewer and then one or both of two control templates can be added to the previewer procedure.

### Fixes and updates

The following lists all changes, fixes and updates we have made to the templates since February 23, 2003.

---

#### March-18-2004:

---

1. Template Added option to all templates to not generate code. This comes in handy when debugging applications. Simply turn off the generation for the templates you don't want to generate.

---

#### March-10-2004:

---

1. Template Added several embed points to the PrinterSelection routines. This allows to set report properties before the report is printed.
2. Template Added option to set the Target() to the report just before calling the previewer. This allows you to reference the report inside the previewer. A ReportRef reference variable is also added when using the Printer Selection templates. Loc:ReportJobName variable added that contains the jobname of the original report.

---

#### March-03-2004:

---

1. Template Verified that it works with Vivid-Help Email Report after the last changes.

---

#### February-27-2004:

---

1. Template Previewer extension was not using the C6 SkipPreview option. Fixed in ABC
2. Template Previewer Export was not using the tagging in the previewer, i.e. it exported all pages, not just tagged pages.

---

#### February-24-2004:

---

1. Documentation Have cleaned up the new features and made some additions to the documentation. Most of the documentation work still not done.

2. Demo Demo added to show the new features.

---

**February-23-2004:**

---

1. New feature Previewer can now execute the generated target file to open it in the associated program.

---

**December-4-2003:**

---

1. New feature Previewer now fully supports Report Targets in both Clarion and ABC templates under Clarion 6.
2. New feature Clarion6 Support extension template. Only available in Clarion 6. Sets up support for exporting reports to PDF, HTML, XML and Text
3. New feature "Save buttons for Clarion 6" control template. Populates 4 buttons to be target buttons. The buttons are unhidden based on how many targets are available.
4. New feature "Save As button for Clarion 6" control template. Populates ONE button that pops up the standard target selection window the same way as the standard previewer does.

---

**November-25-2003:**

---

1. Fix Install now has a C6 dll.
2. Fix When using with Fomin, it would generate an omit statement twice. Fixed.

---

**February-18-2003:**

---

1. Fix PDF-Tools button: If filedialog was cancelled it would still write the pdf file. Fixed.
2. Fix PDF-Tools button: If the global template was not found it would still generate code. Fixed.
3. Fix PDF-Tools button: No tooltip specified. Fixed.

---

**February-17-2003:**

---

1. New feature Added support for PDF-Tools 2.5 as previewer
2. New feature Added control template that prints to PDF from the previewer
3. Fix Changed template code to use Self.PreviewQueue instead of SELF.Preview.ImageQueue This eliminates the need to change the abreport.inc file.
4. Fix Changed install procedure to remove program to change abreport.inc (see 3)

## The Icetips Previewer Editor

The Icetips Previewer Editor enables you to edit the theme files so you can select icons for the Previewer buttons, etc. so they are all correct once you have created the Previewer procedure. This saves you time in going in and picking the icons each time you create a new previewer. You can also set the window background and the background for the pagelist.

The theme files use .itp extension. The first time the Previewer Editor runs, it automatically associates the .itp extension to the Icetips Previewer Editor so you can simply double click on any files with the .itp extension to load them into the Icetips Previewer Editor. That said, the .itp files are simple ini files so you can use any editor that supports pure ascii to edit these files, also an example file is at the end of this section.

We provide you with no less than 89 icons and cursors in 5 different color themes, Orange, Yellow, Blue, Green and Gray for you to pick from and you can, of course, use your own.

*The Previewer Editor attempts to do it's absolute best to conform to the Clarion standard search path for image files. When you pick an icon and you opt to use the filename only, i.e. not get the whole path with it, the Editor will go searching for the appropriate icon.*

*Because you can be editing for any version of Clarion from Clarion 4 to Clarion 5.5 at the same time, the logic we have applied is to look for the most recent version of Clarion first, i.e. C5.5, then C5 and finally C4. We use the Clarion Redirection file for each version to search for the images. The Redirection file can have entries like:*

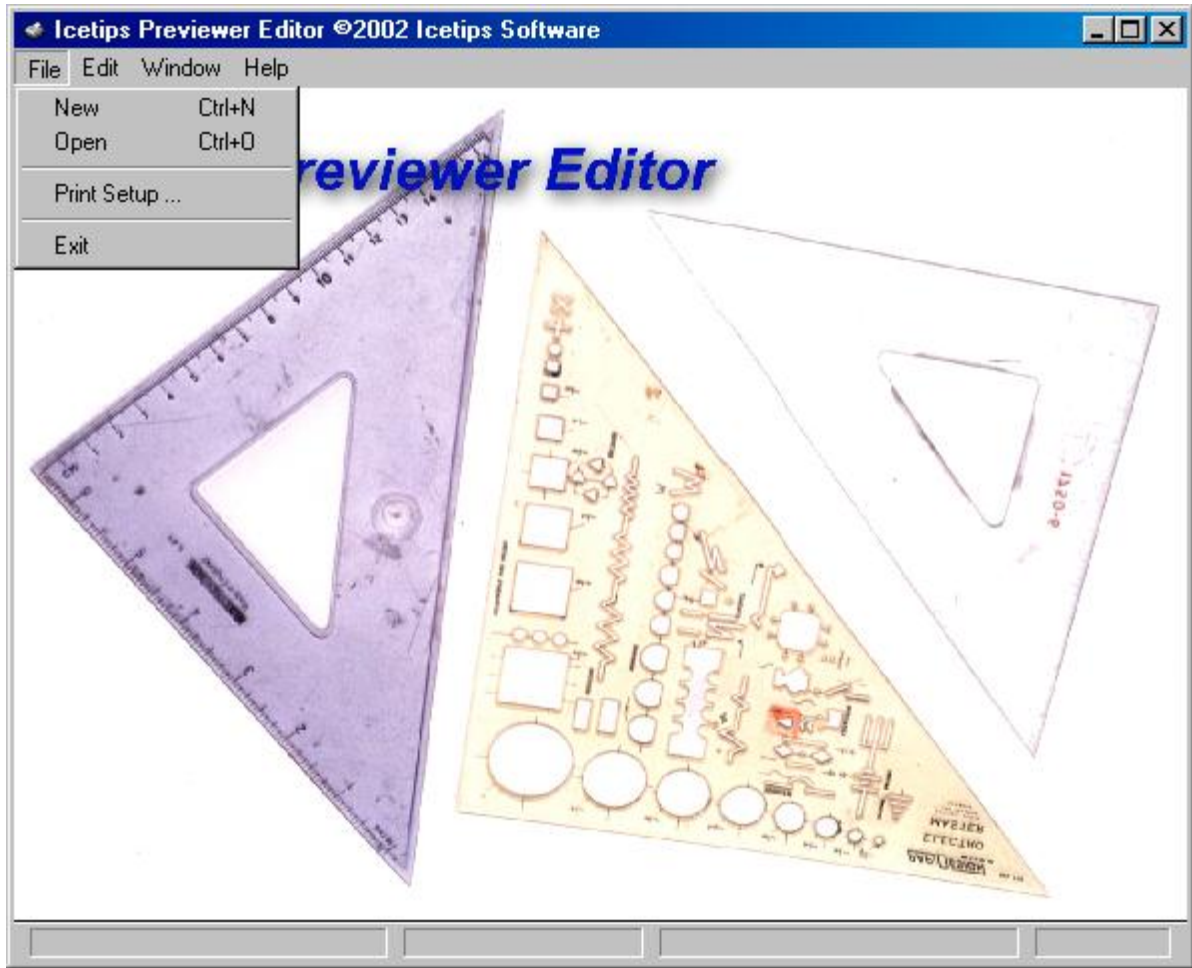
```
*.ico=.;.\z-icons ; %ROOT%\3RDPARTY\Images; %ROOT%\images ;  
%ROOT%\template
```

*This means that we will first look in the current app directory (.), then in \z-icons from the app directory (my personal setup), then in the %ROOT%\3rdParty\Images and so on and so forth. If you have C5.5, C5 and C4 installed, it will start searching based on the C5.5 Redirection file, then the C5 redirection file, then the C4 file.*

*If the image cannot be found in any of these directories, it means that either you mistyped it (if you didn't select it) or the path you selected it from is not in the redirection file, in which case the Clarion linker will not find it either and will be unable to link your application using a Previewer procedure created by this theme file.*

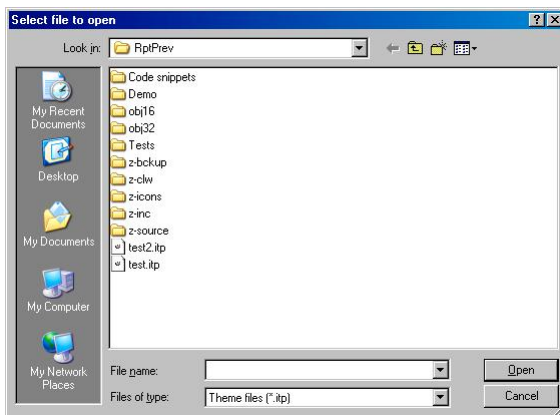
The main menu in the Icetips Previewer Editor allows you to create new files and open existing files from the File menu. In both cases it will ask you for a filename to create or open, unless you have run it by double clicking on an existing file in Windows Explorer. If you are creating a new file, you do not have to type the itp extension, the program will automatically append it to the filename.

## Main window



*The Icetips Previewer Editor Window*

The Previewer Editor uses the Ctrl+N hotkey for New file and Ctrl+O for Open file. When you use the File|Open, you will be prompted for a file to open. Note that the screenshot here is from Windows-XP and will probably be different if you are running other versions of Windows.



*Open File dialog box in Windows XP*

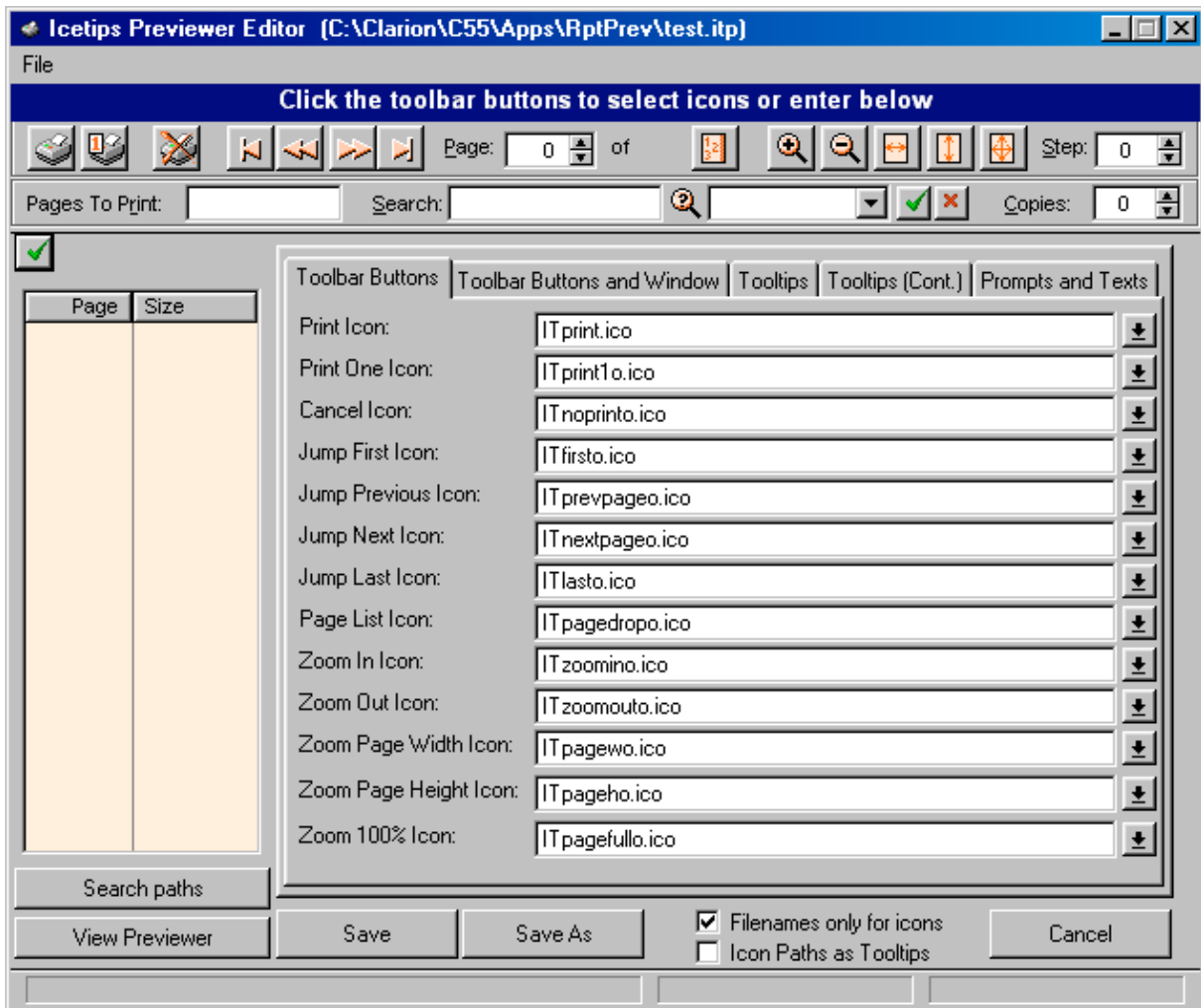


## The Editor window

When the Editor window opens, you will see a toolbar with icons at the top. We have made this window look like the true Previewer, so that you can see how it will look when you use the Wizard to create your previewer using your theme file. On top of that, you can run an actual previewer right from the editor that is fully functional, except it does not print the page.

### Toolbar Buttons tab

The first tab on the main window, "Toolbar Buttons", has entries for all the icons on the top row on the toolbar.



*The Icetips Previewer Editor window*

To select the icon, you can either click on the buttons on the toolbar or the buttons to the right of the icon entry fields.

### Filename only for icons option

The "Filename only for icons" option will strip the path off the icon names when you select them. If this option is unchecked the filename will contain drive and pathname when you select it. This means that the icon path will be hard coded into the Previewer procedure with path and everything - if you move the icon on your hard drive, the linker will not be able to find it any more!

### Icon Paths as Tooltips

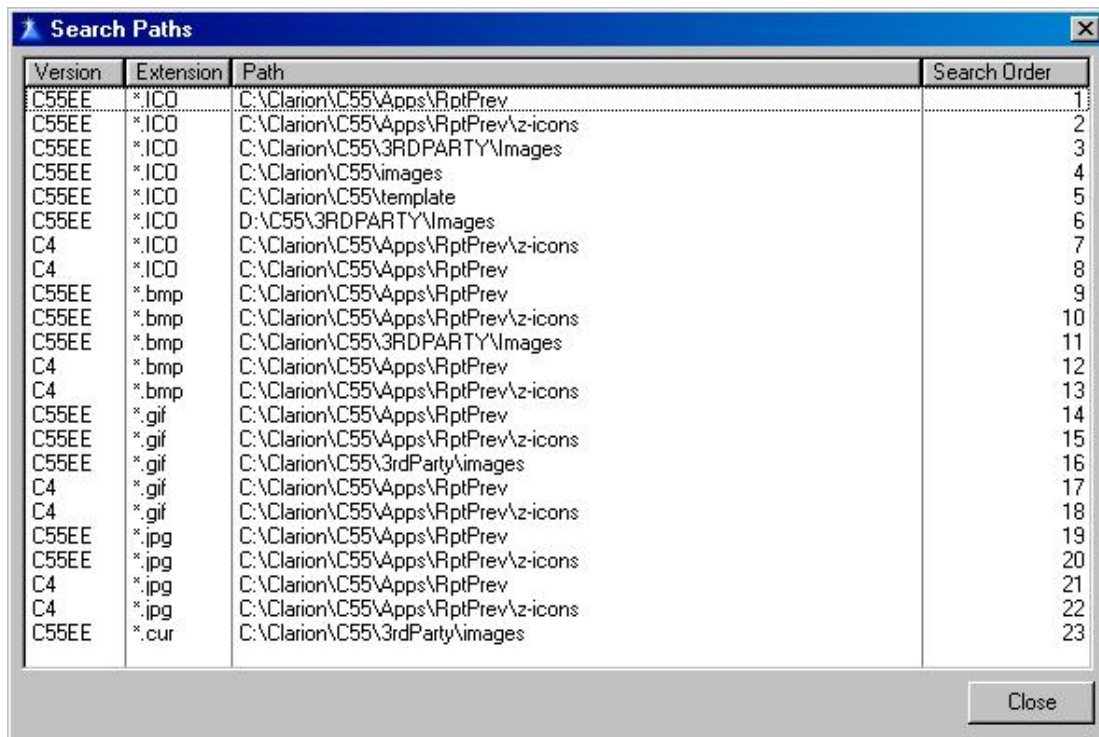
If this is checked, the buttons in the toolbar will show the path and filename of the icon that is assigned to that button in the tooltip. If this is unchecked it will show the tooltip assigned to the button as tooltip text.

We suggest you play around in this editor for a while to get familiar with how it works and what icon is what.

If you click the Cancel button you will lose your changes, but you will be warned so you can go back and hit the Save button instead.

### Search Path button

The Search Path button simply allows you to view the search paths that the program will use to find icons and other images. You can use .ico, .bmp, .gif or .jpg files as "icons" in the previewer, so it will show you the search paths for all the image types.



The Search Paths window, showing search paths for C5.5 and Clarion 4.

When the Editor searches for the files for you, it will start at the first entry for the correct file extension, for example, if it's looking for .gif files, it will start on entry number 14 in the list

and work down to entry number 18. If the Editor can't find it, the button will show up without any icon, so you should notice if your icon is not found.

### View Previewer button

The View Previewer button allows you to see how the previewer will look using your theme file. When you click on the View Previewer button, the program prompts you to save the changes you have made so they will show up in the Previewer. If you do not save the changes, the Previewer will show you how the theme file was before you made any changes.

### Toolbar Buttons and Window tab

The second tab, "Toolbar Buttons and Window", contains entries for the three buttons on the second row on the toolbar and the "Print this Page" toggle button which is on the window, top left corner.

### Flat Buttons option

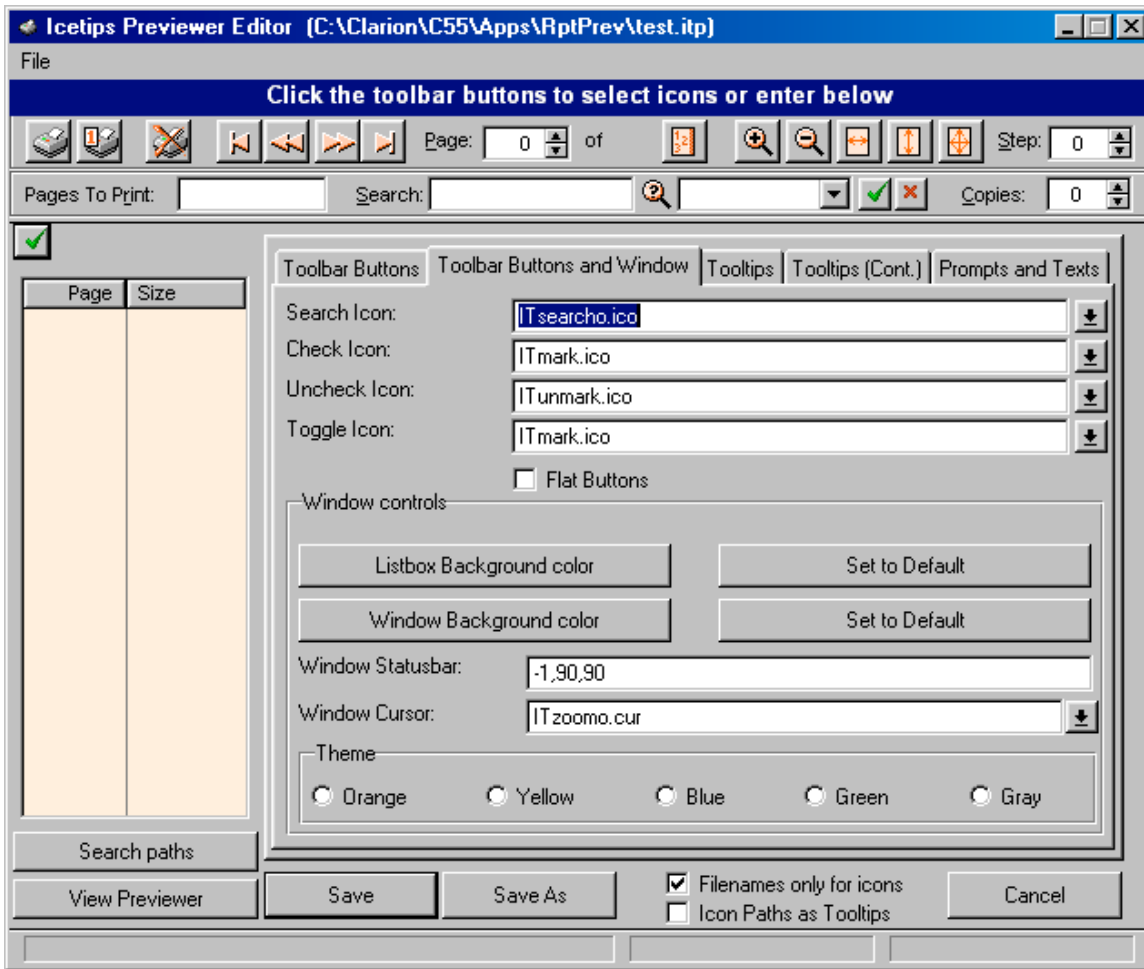
The Flat Buttons option will set all the buttons either flat or not flat, except the Search button, which is flat by default because of it's location.

### Previewer Window and pagelist background

The second tab also allows you to set up the window background color and the page list background color. The "Set to Default" buttons will reset the window background color back to `COLOR:BtnFace` and the listbox background color to `ODCEDFEH` which is a pale tan color.

### Previewer Window statusbar

The window statusbar entry allows you to set up the statusbar for the window. By default it is set up to use -1,90,90 which means that there are two 90 dialog unit sections to the right of an adjusting section, which follows the window size. The Previewer uses 3 sections in it's code, so there should be 3 sections there unless you plan on changing the previewer code to accommodate fewer sections.



The Toolbar Buttons and Windows in The Icetips Previewer Editor

## Cursor

The Window Cursor allows you to select the cursor for the window, which is used when the mouse cursor is hovering over the page image. The default cursor is from the orange theme. It is a little magnifying glass with a plus and minus in it. There is one cursor file provided for each color theme, named ITCursor?.cur, where the ? is o for Orange, y for Yellow, g for Green, b for Blue and r for Gray.

## Theme option

The Theme option allows you to reset all the icons to a particular scheme. Note that this will override any changes or settings that you have made to the icons and the cursor selections. If you click one of the theme options, you will be asked if you want to override any of the options.

## Save As

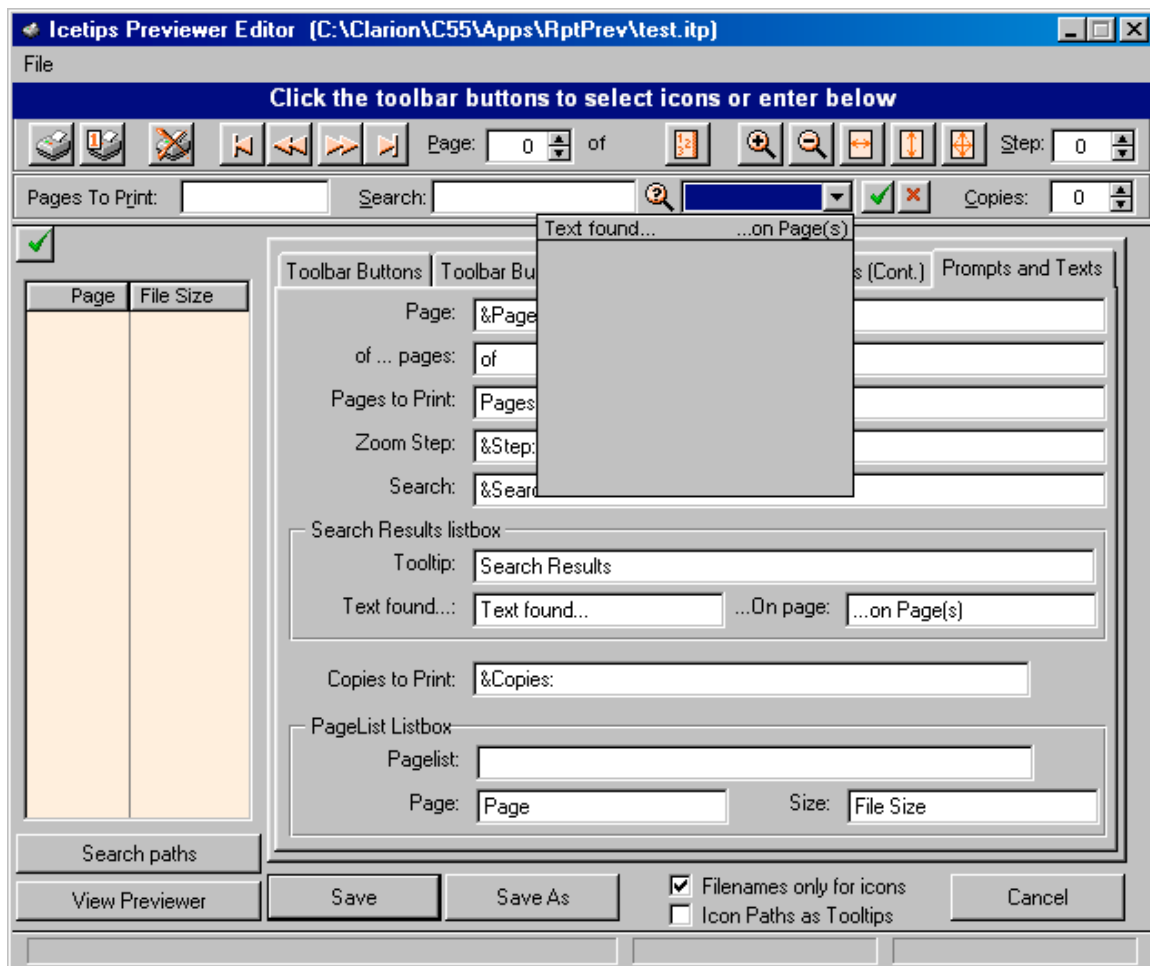
The Save As button prompts to you save the current setup in a new itp file. Again, if you do not type in the extension, the program will add it automatically. Note that if you pick an

existing filename the program will immediately overwrite it - i.e. the file is saved immediately to the new name.

## Tooltips + Prompts and Texts

On the two tooltips tabs you can set the texts for the tooltips of every control that has a tooltip in the Previewer. This along with the Prompts and Texts tab allows you for example to create a theme that is translated to a different language. There are a few additional places where texts that appear on the window are set in code, but most of the text properties of the Previewer window can be changed here.

The tooltips and the texts update automatically on the window immediately when each entry field is accepted, so you will see immediately how your changes will fit on the generated previewer window. This includes the headers on the search results drop down and the headers on the PageList.



*Immediate visual update of changes to tooltips, prompts, texts and listbox headers*

This new capability in version 1.001 allows you to create theme files with different languages, so you can easily create Previewer procedures in the language you want.

## The Search Path window

The Search Path window allows you simply to view the order in which the Previewer Editor searches for image files, and it should be the same sort order as Clarion uses. Because of the multi-version compatibility in the Icetips Previewer, it is possible that it may find icons from more recent version before it finds the one for a previous version, which may be the one you are working with.

## The .itp file format

The itp files are simple ini files. You can use any program to edit the files that can edit and save ascii files.

*If you at any time attach files of this type to emails, it may cause problems when you try to use them. The reason is that when text files are sent by email they are usually reformatted to unix text files, which means that the CR+LF pair is reduced to a single LF character. If you need to send itp files in emails, use WinZip or similar products to create a zip file with the itp files and then email the zip file.*

```
[Icons]
; Print icon does not change with themes
ITPPrintIcon=ITprint.ico
ITPPrintOneIcon=ITprintlg.ico
ITPPrintCancelIcon=ITnoprintg.ico
ITPJumpFirstIcon=ITfirstg.ico
ITPJumpPrevIcon=ITprevpageg.ico
ITPJumpNextIcon=ITnextpageg.ico
ITPJumpLastIcon=ITlastg.ico
ITPPageListIcon=ITpagedropg.ico
ITPZoomInIcon=ITzooming.ico
ITPZoomOutIcon=ITzoomoutg.ico
ITPZoomPageWIcon=ITpagewg.ico
ITPZoomPageHIcon=ITpagehg.ico
ITPZoomPage100Icon=ITpagefullg.ico
ITPSearchIcon=ITsearchg.ico
; The test of the icons do not change with themes
ITPCheckIcon=ITmark.ico
ITPUnCheckIcon=ITunmark.ico
ITPToggleIcon=ITmark.ico
```

```
[Colors]
Listbox back=14478846
Window back=12632256
```

```
[Window]
Statusbars=-1,90,90
```

```
[Buttons]
Flat Buttons=0
```

```
[Cursor]
ITPZoomCursor=ITzoomo.cur
```

```
; The Tooltips section was added in version 1.001
[Tooltips]
```

```
ITPPrintAllButtonTip=Print the report
ITPPrintOneButtonTip=Print current page only
ITPCancelButtonTip=Exit without printing
ITPJumpFirstButtonTip=Jump to First page
ITPJumpPrevButtonTip=Jump to Previous page
ITPJumpNextButtonTip=Jump to Next page
ITPJumpLastButtonTip=Jump to Last page
ITPCurrentPageTip=Select page to display
ITPShowPageListTip=Show pagelist
ITPZoomInButtonTip=Zoom in
ITPZoomOutButtonTip=Zoom out
ITPZoomToPageWidthTip=Zoom to page Width
ITPZoomToPageHeightTip=Zoom to page Height
ITPZoom100PercentTip=Zoom 100%
ITPZoomStepValueTip=Steps to Zoom
ITPPagesToPrintTip=Valid options: All, None, Even, Odd, individual pages
(1,2,5) and page range (1-5)
ITPSearchStringTip=Enter text to search for
ITPSearchButtonTip=Search
ITPTagAllFoundTip=Tag all pages with text to be printed
ITPUntagAllFoundTip=Tag all pages without text to be printed
ITPCopiesToPrintTip=Enter Copies to print
ITPTogglePrintButtonTip=Toggle Print/Not print this page
ITPPagePromptText=&Page:
ITPOfStringText=of
ITPPagesToPrintPrompts=Pages To P&rint:
ITPZoomValuePrompts=&Step:
ITPSearchStringPrompts=&Search:
ITPFoundPageTip=Search Results
ITPFoundPageHeader1=Text found...
ITPFoundPageHeader2=...on Page
ITPCopiesToPrintPrompts=&Copies:
ITPList2Tip=
ITPList2Header1=Page
ITPList2Header2=Size
```

*The .itp file format is registered as ITPFile in the registry. There are two entries in the HKEY\_CLASSES\_ROOT to implement the association and the key values are as follows:*

```
HKEY_CLASSES_ROOT\.itp\default = ITPFile
HKEY_CLASSES_ROOT\ITPfile\shell\open\command\default =
C:\Clarion\C55\3rdParty\Tools\ITPrev\ITPrevEd.exe %1
```



## Compatibility & technological issues

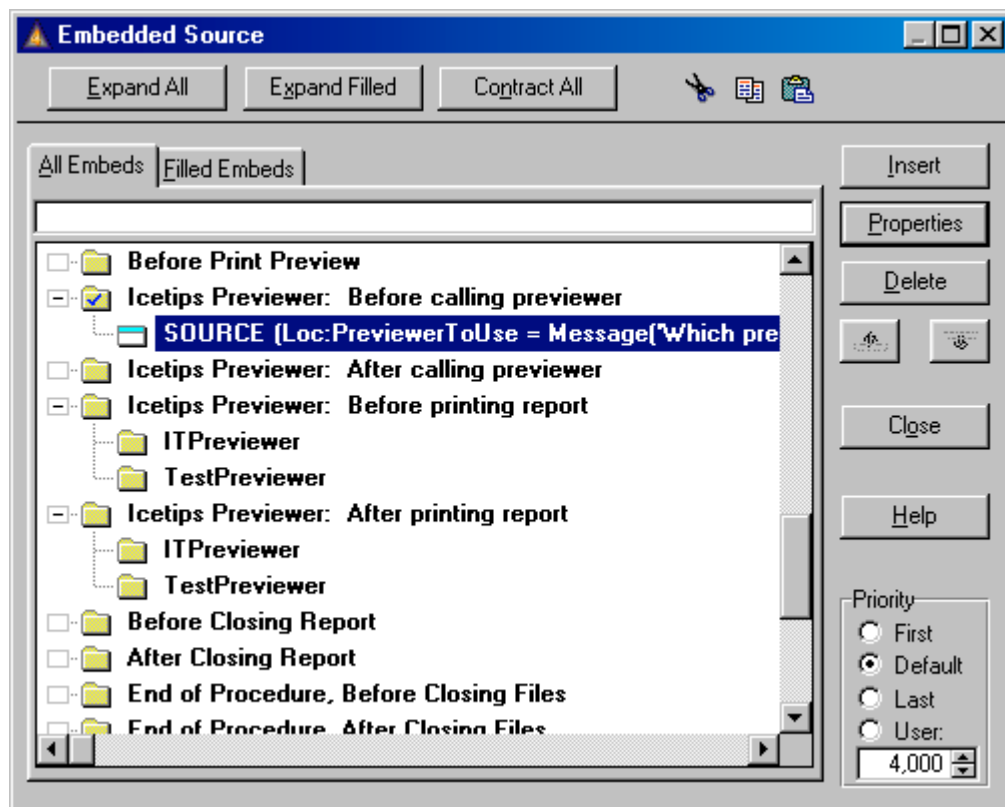
Although there are a few things to consider when using the Icetips Previewer, it is easy to use and we have done everything we can to make it so.

### Compatibility with earlier versions of Clarion

While we try to maintain backward compatibility as much as we can, our customer base is mainly using Clarion 5.5 and Clarion 6.0 and those two versions will get our primary attention in development and testing. If you find that there are problems using the Icetips Previewer in Clarion 4 or Clarion 5, apart from what is noted here, we would like to know about them so we can look into the problem and hopefully fix it.

### Clarion 4, ABC

Clarion 4, ABC, presented us with a strange challenge in the way that it would not under any circumstances import the TXA correctly. Even if all the code was there in the embeds, it was not generated at all and some standard code that should have been generated was simply left out. The "fix" was simply to import the Legacy TXA into the ABC window procedure and it works fine, BUT it means that you have Legacy embed points in the otherwise ABC structured procedure, which may be confusing.



*The Icetips Previewer embeds in Clarion 4, Legacy application*

## Optionally using the Clarion or CPCS standard previewers

In order to use the Clarion or CPCS Standard previewer in Legacy and ABC, you need to handcode it as it is never called after you apply the Icetips Previewer Extension template to your report.

In Legacy this requires a few lines of code that should go into Icetips Previewer: Before calling previewer embed. Example:

```
If Loc:PreviewerToUse = 3
  ReportPreview(PrintPreviewQueue)
  If GlobalResponse = RequestCompleted
    MyReport{PROP:FlushPreview} = True
  End
End
```

In ABC the code goes into the same embed - Icetips Previewer: Before calling previewer:

```
If Loc:PreviewerToUse = 3
  PARENT.AskPreview
End
```

Note that depending on what version of Clarion you are using, the embeds may look a little bit different from the images you see in this document. The Icetips Previewer embeds are always called the same, but depending on what version you are using, and also if it is an ABC, Legacy or CPCS report, the embeds may be in a different order. For example in C4, Legacy, the Before calling previewer and After calling previewer embeds are both above the Before printing report embeds.

## Clarion 5, Legacy

This combination caused us quite a headache for version 1.001. Minor changes in the appearance and generation order between Clarion 4, Clarion 5 and Clarion 5.5 caused the Legacy Extension template to generate code out of correct order. This was only obvious if code was put into the Before Print Preview embed as this code would generate inside the If structures generated by the Extension template causing weird behaviour of the report - details printed in this embed would appear on a page "after" the report etc. After investigating this thoroughly, we discovered that there are some differences in how Clarion 5 interpreted priorities and also how it build the embed tree. None of the Icetips embeds showed up in Clarion 5, Legacy! The downside of this is that the Clarion 5 Extension template does not work at all in Clarion 4 or Clarion 5.5, so we are forced to supply the Clarion 5 template as a separate extension file and then copy the correct one to the right name during the installation. No problems have been reported with neither Clarion 4 or Clarion 5.5 nor with the ABC templates in any version.

## Other Clarion third party products

In the first two years of it's lifetime the Icetips Previewer has been tested pretty extensively with other third party products. It is compatible with several report generating products, such as CPCS (<http://www.cpcs-inc.com>), Fomin Report Builder (<http://www.fomintools.com>), PDFTools (<http://www.docu-track.com>), EmailReport from VividHelp (<http://www.vividhelp.com>), Icetips Xplore (<http://www.icetips.com/xplore.php>), EasiListPrint from Ingasoft (<http://www.ingasoftplus.com/id74.htm>) and most recently the wPDFControl Wrapper from Klarisoft ([http://www.klarisoft.com/KSNews\\_127.htm](http://www.klarisoft.com/KSNews_127.htm)). If you find any problems using it with your other tools, then we would appreciate it if you would let us know. We will do everything we can to resolve the problem as soon as possible.

The original release of the Icetips Previewer was mostly developed in Clarion 4 build B and Clarion 5.5 Enterprise Edition, build E. Version 2.0 was mostly developed and tested in Clarion 6.0 and also tested in Clarion 5.5H. While we consider the previewer backward compatible with C5 and C4, we will not be supporting those versions specifically.

The installation is built with LinderSoft SetupBuilder version 4.1.

## Demos

The Icetips Previewer comes with a few demos that you can use as a guide. We will now briefly discuss those demos and also how you can add functionality to them.

### Old demos

We supply the original Clarion 4 created demos that should work with all versions of Clarion, from 4 to 6. These are called ABCDemo.app and ITPDemo.app and they use the ITPDemo.dct dictionary. We have not modified these demos in version 2.0

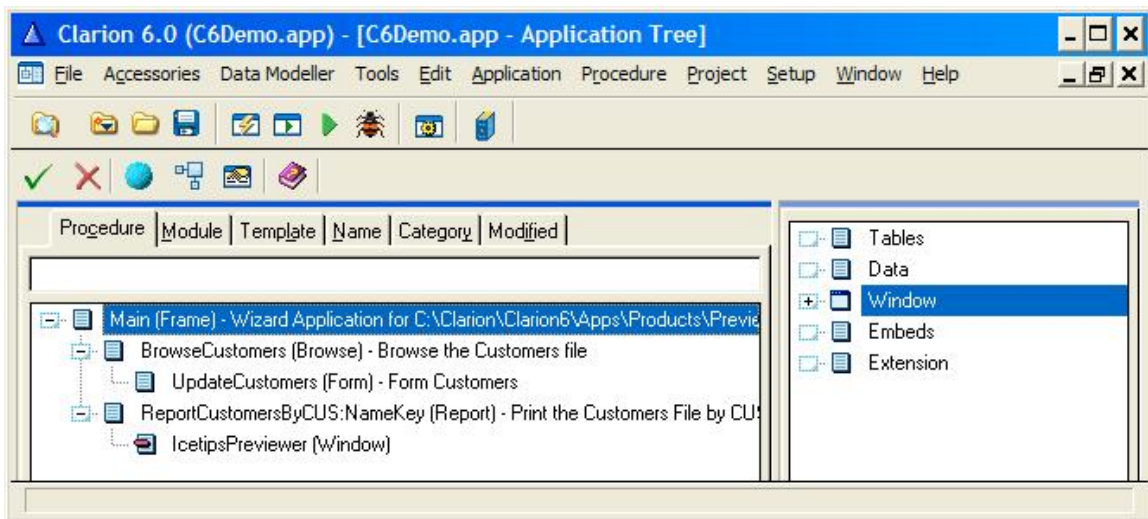
### New demos

#### Clarion 6 demo

The Clarion 6 demo, called C6Demo.app uses the ITPC6Demo.dct dictionary, so they are not mixed up with the older demos. The Clarion 6 demo in reality is just plain C6 app with the Icetips previewer in it. Why? Because even if you own Clarion 6, you may not have all the target templates, so we decided NOT to include them, but rather go through here how you add the targets you want and how you implement them in your Icetips Previewer.

When you open the C6Demo.app you will see a plain wizard generated application with the Icetips Previewer. Nothing has been added to it. Please note that in this section we are teaching you how to implement the target templates from Softvelocity. If you do not have those templates, you need to contact Softvelocity (<http://www.softvelocity.com>) or their distributors about purchasing them. We do not sell them and Icetips Software can not help you obtaining these templates.

### Adding target support to your Clarion 6 application and Icetips Previewer



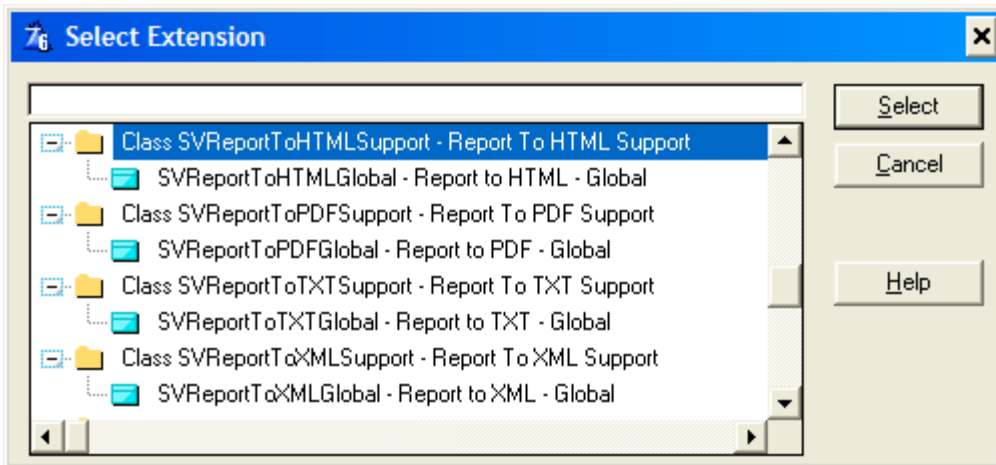
The first thing we need to do is to register the target templates.

Registering the templates

1. Close your C6Demo.app
2. Select "Setup|Template Registry" from the Clarion 6 IDE main menu
3. Click the "Register" button
4. Select the RTAR\*.TPL files that you have. Which template files you have depends on if you are using the Enterprise Edition or the Professional Edition. The Professional Edition does NOT include the target templates but you can purchase them separately from Softvelocity (<http://www.softvelocity.com>)
5. Click the "Open" button on the "Template File" window.
6. The templates you selected will now be registered

Adding Target templates to your application

7. Make a backup copy of your C6Demo.app and put it in a safe place or work with the copy. We will presume that you are working with an app named C6Demo.app.
8. Open your C6Demo.app in Clarion 6.
9. Select "Application|Global Properties" from the Clarion 6 IDE main menu.
10. Click on the "Extensions" button to open the Global Extensions. There should be no extension templates present at this time.
11. Click on the "Insert" button to add the global extension templates.



12. Select the target template you want to add.
13. Click the "Select" button on the "Select Extension" window.
14. Repeat step 11 to 13 for all the target templates you want to add.

Add the Icetips Previewer Global Extension template

15. Click on the "Insert" button to add the Icetips Previewer "Enable Printer Selection" Global extension - see page 28 for more information.
16. Check the "Prototype ShellExecute for C6 target classes"
17. Click the "Ok" button to go back to the Global properties window
18. Click the "Ok" button to go back to the application tree.

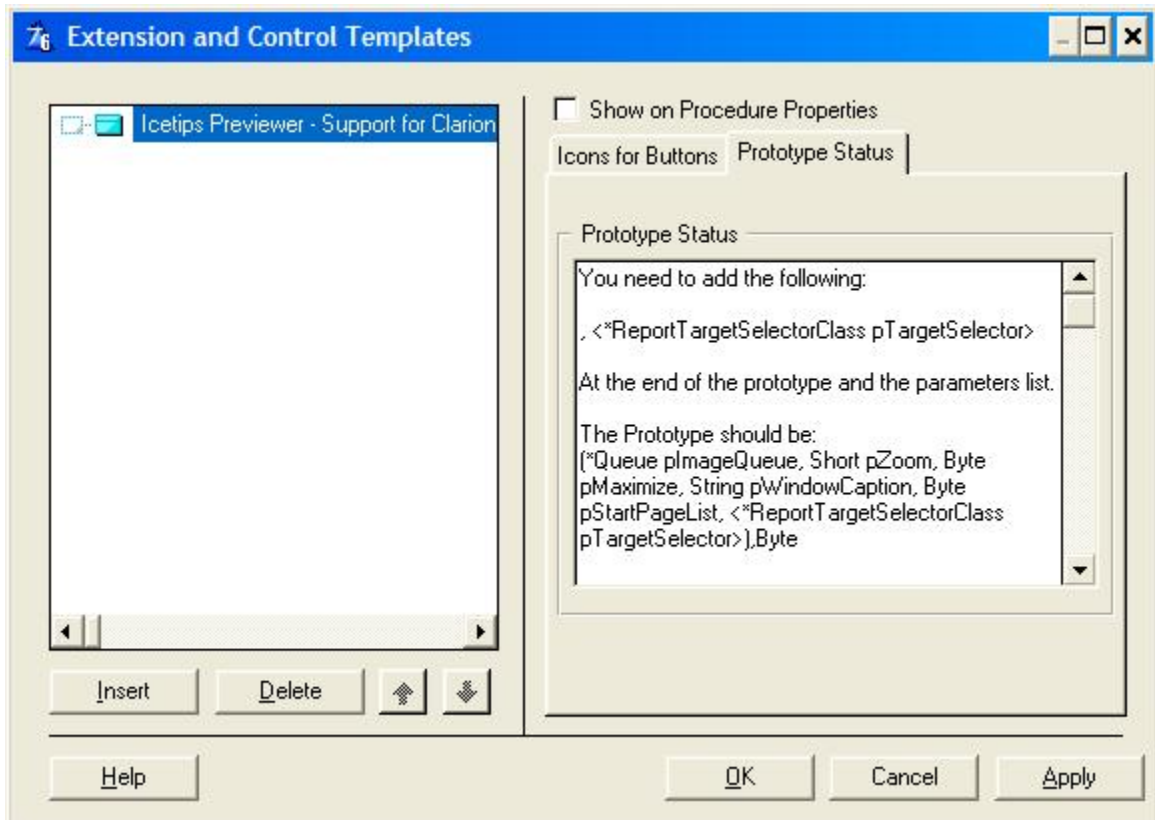
Add Clarion 6 support to the Icetips Previewer procedure

19. Double click on the IcetipsPreviewer procedure.
20. Click the "Extensions" button.

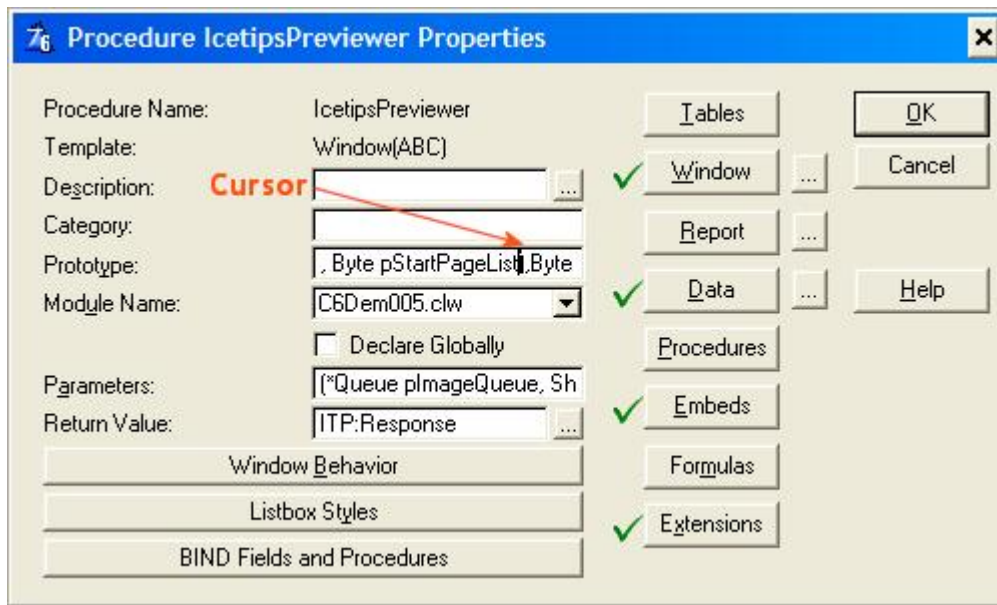
21. Click the "Insert" button to add the Clarion 6 support template.
22. Locate the "IcetipsPreviewer" class and then select the "ITPClarion6Support - Icetips Previewer - Support for Clarion 6" extension template from the list.

Modify the Icetips Previewer Prototype

23. Click the "Prototype Status" tab
24. You will now see a warning about the prototype not being correct.



25. Copy the ", <\*ReportTargetSelectorClass pTargetSelector>" text from the textbox using Ctrl-C
26. Click the "Ok" button to go back to the Procedure Properties window
27. Click in the "Prototype" entry field
28. Press the End key on the keyboard to go to the end of the line
29. Place the cursor between the T at the end of "pStartPageList" and the closing parenthesis )



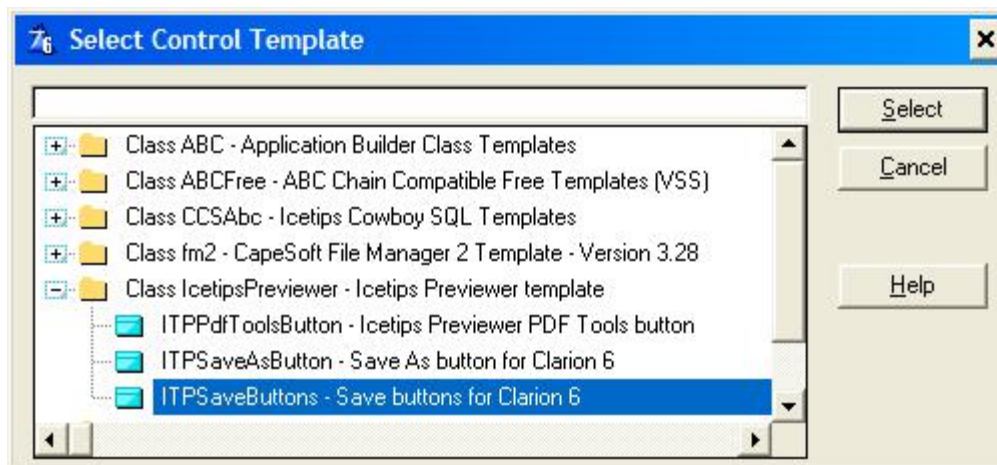
30. Paste the text on the clipboard, ", <\*ReportTargetSelectorClass pTargetSelector>" into the prototype at the cursor position. The resulting prototype should look like this:

```
(*Queue pImageQueue, Short pZoom, Byte pMaximize, String pWindowCaption,
Byte pStartPageList, <*ReportTargetSelectorClass pTargetSelector>),Byte
```

- 31. Click in the "Parameters" entry field and repeat steps 28-30
- 32. Click the "Extensions" button.
- 33. Click the "Prototype Status" tab on the "Icetips Previewer - Support for Clarion 6" and it should now state that the prototype appears to be correct.

Add Target buttons to the Icetips Previewer

- 34. Click the "Window" button to open the Icetips Previewer in the Window formatter.
- 35. Select "Populate|Control Template..." from the Window formatter main menu.
- 36. Locate the "Icetips Previewer" class and select the "ITPSaveButtons - Save buttons for Clarion 6" from the list of control templates.





37. Position the mouse cursor few pixels to the right of the group with the "Step" prompt on the Previewer toolbar.
38. Click with the mouse to drop the control template on the toolbar.



Set Target button options

39. Right click on any of the new button and select "Actions" from the popup menu.
40. Check the "Retrieve target filename" checkbox. The "Open target file in associated program" checkbox should now be enabled. If not, check that you checked the "Prototype ShellExecute for C6 target classes" in step 16
41. Check the "Open target file in associated program" checkbox.
42. Click the "Ok" button to bring you back to the window formatter
43. Select "Exit" from the Window formatter menu to take you back to the Procedure properties window.

Compile and run

44. Compile and run the C6Demo.app. If you get errors, check to make sure that you changed BOTH the prototype and the parameters on the IctipsPreviewer procedure.

You also have this application in the 3rdParty\Examples\ITPrev directory, called C6DemoPDF.app

## Installed files

The Icetips Previewer installs quite a few files as it includes over 100 icons and cursor files. Following is a list of all files that are installed as of version 2.0 releases in March, 2004.

This list may not be absolutely complete as some of the files are still subject to change at the end of the documentation phase. Filesizes and date/time stamps may also change before release. %ROOT% refers to your Clarion installation root directory

### Files in: %ROOT%\3rdParty\Docs\ITPrev

| Date       | Time     | Size      | Filename       |
|------------|----------|-----------|----------------|
| 03/19/2004 | 02:06 PM | 1,058,403 | ITPrev.pdf     |
| 03/19/2004 | 01:28 PM | 6,067     | ITPreviwer.txt |

### Files in: %ROOT%\3rdParty\Examples\ITPrev

| Date       | Time     | Size      | Filename      |
|------------|----------|-----------|---------------|
| 01/23/2002 | 04:00 PM | 282,112   | ABCDemo.app   |
| 02/24/2004 | 06:54 PM | 215,808   | C6Demo.app    |
| 01/23/2002 | 05:10 PM | 298,240   | customers.tps |
| 12/10/1997 | 05:13 PM | 338       | FINGER.CUR    |
| 02/21/2004 | 04:33 PM | 2,048     | ITPC6Demo.dct |
| 01/23/2002 | 04:08 PM | 276,480   | ITPDemo.app   |
| 01/23/2002 | 02:46 PM | 1,792     | ITPDemo.dct   |
| 09/25/2001 | 03:00 PM | 20,454    | itwiz.gif     |
| 01/23/2002 | 02:19 PM | 1,440,054 | Peace.bmp     |

### Files in: %ROOT%\3rdParty\Images

| Date       | Time     | Size  | Filename       |
|------------|----------|-------|----------------|
| 03/11/2003 | 01:15 PM | 2,238 | EXP_CSV.ico    |
| 03/11/2003 | 01:10 PM | 2,238 | EXP_DOTM.ico   |
| 12/01/2003 | 04:25 PM | 5,494 | EXP_HTM.ico    |
| 12/01/2003 | 08:01 PM | 3,262 | EXP_HTMsm.ico  |
| 03/11/2003 | 12:33 PM | 2,238 | EXP_PDF.ico    |
| 12/01/2003 | 08:01 PM | 3,262 | EXP_PDFsm.ico  |
| 03/11/2003 | 11:58 AM | 2,238 | EXP_PRI.ico    |
| 03/11/2003 | 01:21 PM | 2,238 | EXP_RTF.ico    |
| 03/11/2003 | 12:26 PM | 2,238 | EXP_TXT.ico    |
| 12/01/2003 | 07:53 PM | 3,262 | EXP_TXTsm.ico  |
| 03/11/2003 | 01:05 PM | 2,238 | EXP_XLS.ico    |
| 03/11/2003 | 01:29 PM | 2,238 | EXP_XML.ico    |
| 12/01/2003 | 07:57 PM | 3,262 | EXP_XMLsm.ico  |
| 01/11/2002 | 06:16 PM | 2,238 | ITfirst.ico    |
| 01/17/2002 | 12:05 AM | 2,238 | ITfirstb.ico   |
| 01/18/2002 | 12:02 PM | 2,238 | ITfirstg.ico   |
| 01/11/2002 | 06:16 PM | 2,238 | ITfirsto.ico   |
| 01/18/2002 | 02:52 PM | 2,238 | ITfirststr.ico |
| 01/18/2002 | 01:05 PM | 2,238 | ITfirsty.ico   |
| 01/11/2002 | 06:17 PM | 2,238 | ITlast.ico     |
| 01/17/2002 | 12:07 AM | 2,238 | ITlastb.ico    |
| 01/18/2002 | 12:04 PM | 2,238 | ITlastg.ico    |

|            |          |       |                  |
|------------|----------|-------|------------------|
| 01/11/2002 | 06:17 PM | 2,238 | ITlasto.ico      |
| 01/18/2002 | 02:53 PM | 2,238 | ITlastr.ico      |
| 01/18/2002 | 01:06 PM | 2,238 | ITlasty.ico      |
| 01/11/2002 | 06:56 PM | 2,998 | ITmark.ico       |
| 01/11/2002 | 06:20 PM | 2,238 | ITnextpage.ico   |
| 01/17/2002 | 12:07 AM | 2,238 | ITnextpageb.ico  |
| 01/18/2002 | 12:04 PM | 2,238 | ITnextpageg.ico  |
| 01/11/2002 | 06:20 PM | 2,238 | ITnextpageo.ico  |
| 01/18/2002 | 02:54 PM | 2,238 | ITnextpager.ico  |
| 01/18/2002 | 01:06 PM | 2,238 | ITnextpagey.ico  |
| 01/18/2002 | 12:21 PM | 2,238 | ITnoprintb.ico   |
| 01/18/2002 | 12:23 PM | 2,238 | ITnoprintg.ico   |
| 01/18/2002 | 03:10 PM | 2,238 | ITnoprinto.ico   |
| 01/18/2002 | 02:56 PM | 2,238 | ITnoprintr.ico   |
| 01/18/2002 | 01:08 PM | 2,238 | ITnoprinty.ico   |
| 01/18/2002 | 03:14 PM | 2,238 | ITpageb.ico      |
| 01/12/2002 | 12:12 PM | 2,238 | ITpagedrop.ico   |
| 01/16/2002 | 11:59 PM | 2,238 | ITpagedropb.ico  |
| 01/18/2002 | 12:06 PM | 2,238 | ITpagedroprg.ico |
| 01/12/2002 | 12:12 PM | 2,238 | ITpagedropo.ico  |
| 01/18/2002 | 02:58 PM | 2,238 | ITpagedropr.ico  |
| 01/18/2002 | 01:10 PM | 2,238 | ITpagedropy.ico  |
| 01/12/2002 | 11:54 AM | 2,238 | ITpagefull.ico   |
| 01/17/2002 | 12:00 AM | 2,238 | ITpagefullb.ico  |
| 01/18/2002 | 12:08 PM | 2,238 | ITpagefullg.ico  |
| 01/12/2002 | 11:54 AM | 2,238 | ITpagefullo.ico  |
| 01/18/2002 | 02:59 PM | 2,238 | ITpagefullr.ico  |
| 01/18/2002 | 01:12 PM | 2,238 | ITpagefully.ico  |
| 01/18/2002 | 03:16 PM | 2,238 | ITpageg.ico      |
| 01/11/2002 | 06:32 PM | 2,238 | ITpageh.ico      |
| 01/17/2002 | 12:00 AM | 2,238 | ITpagehb.ico     |
| 01/18/2002 | 12:08 PM | 2,238 | ITpagehg.ico     |
| 01/11/2002 | 06:32 PM | 2,238 | ITpageho.ico     |
| 01/18/2002 | 02:59 PM | 2,238 | ITpagehr.ico     |
| 01/18/2002 | 01:12 PM | 2,238 | ITpagehy.ico     |
| 01/18/2002 | 03:13 PM | 2,238 | ITpageo.ico      |
| 01/18/2002 | 03:17 PM | 2,238 | ITpager.ico      |
| 01/11/2002 | 06:32 PM | 2,238 | ITpagew.ico      |
| 01/17/2002 | 12:01 AM | 2,238 | ITpagewb.ico     |
| 01/18/2002 | 12:09 PM | 2,238 | ITpagewg.ico     |
| 01/11/2002 | 06:32 PM | 2,238 | ITpagewo.ico     |
| 01/18/2002 | 03:00 PM | 2,238 | ITpagewr.ico     |
| 01/18/2002 | 01:13 PM | 2,238 | ITpagewy.ico     |
| 01/18/2002 | 03:18 PM | 2,238 | ITpagey.ico      |
| 02/17/2003 | 01:58 PM | 2,238 | ITPdfTools.ico   |
| 01/11/2002 | 06:21 PM | 2,238 | ITprevpageb.ico  |
| 01/17/2002 | 12:08 AM | 2,238 | ITprevpageg.ico  |
| 01/18/2002 | 12:10 PM | 2,238 | ITprevpageo.ico  |
| 01/11/2002 | 06:21 PM | 2,238 | ITprevpager.ico  |
| 01/18/2002 | 03:00 PM | 2,238 | ITprevpagey.ico  |
| 01/18/2002 | 01:14 PM | 2,238 | ITprint.ico      |
| 01/18/2002 | 03:20 PM | 2,238 | ITprintb.ico     |
| 01/18/2002 | 12:17 PM | 2,238 | ITprintlg.ico    |
| 01/18/2002 | 12:16 PM | 2,238 | ITprintlo.ico    |
| 01/18/2002 | 12:17 PM | 2,238 | ITprintlr.ico    |
| 01/18/2002 | 03:01 PM | 2,238 | ITprintly.ico    |
| 01/18/2002 | 01:15 PM | 2,238 | ITsearchb.ico    |
| 01/11/2002 | 06:47 PM | 2,238 | ITsearchg.ico    |
| 01/18/2002 | 12:14 PM | 2,238 | ITsearcho.ico    |
| 01/18/2002 | 12:12 PM | 2,238 | ITsearchr.ico    |
| 01/11/2002 | 06:47 PM | 2,238 | ITsearchr.ico    |
| 01/18/2002 | 03:02 PM | 2,238 | ITsearchr.ico    |

```

01/18/2002 01:16 PM          2,238 ITsearchy.ico
01/11/2002 07:00 PM          2,238 ITunmark.ico
08/29/2001 11:06 PM        20,454 itwiz.gif
01/17/2002 12:11 AM          2,238 ITzomb.cur
01/22/2002 10:39 AM          2,238 ITzoomg.cur
01/11/2002 06:37 PM          2,238 ITzoomin.ico
01/17/2002 12:03 AM          2,238 ITzoominb.ico
01/18/2002 12:59 PM          2,238 ITzooming.ico
01/11/2002 06:37 PM          2,238 ITzoomino.ico
01/18/2002 03:03 PM          2,238 ITzoominr.ico
01/18/2002 01:17 PM          2,238 ITzoominy.ico
01/11/2002 06:42 PM          2,238 ITzoomo.cur
01/11/2002 06:12 PM          2,238 ITzoomout.ico
01/17/2002 12:03 AM          2,238 ITzoomoutb.ico
01/18/2002 01:03 PM          2,238 ITzoomoutg.ico
01/11/2002 06:12 PM          2,238 ITzoomouto.ico
01/18/2002 03:04 PM          2,238 ITzoomoutr.ico
01/18/2002 01:18 PM          2,238 ITzoomouty.ico
01/22/2002 10:51 AM          2,238 ITzoomr.cur
01/22/2002 10:43 AM          2,238 ITzoomy.cur

```

#### Files in: %ROOT%\3rdParty\Template

| Date       | Time     | Size   | Filename      |
|------------|----------|--------|---------------|
| 02/27/2004 | 02:33 PM | 43,012 | ITPEXT.TPW    |
| 01/18/2002 | 04:16 PM | 215    | ITPreview.tpl |
| 02/21/2004 | 04:48 PM | 94,698 | ITPWIZ.TPW    |
| 08/29/2001 | 11:06 PM | 20,454 | itwiz.gif     |

#### Files in: %ROOT%\3rdParty\Tools\ITInstall

| Date       | Time     | Size    | Filename      |
|------------|----------|---------|---------------|
| 01/23/2002 | 07:45 PM | 596     | itpScript.itc |
| 02/04/2003 | 11:41 AM | 791,552 | itutil.exe    |

#### Files in: %ROOT%\3rdParty\Tools\ITPrev

| Date       | Time     | Size      | Filename     |
|------------|----------|-----------|--------------|
| 02/05/2002 | 06:06 PM | 4,475,392 | itpreved.exe |

#### Files in: %ROOT%\Bin

| Date       | Time     | Size  | Filename     |
|------------|----------|-------|--------------|
| 01/23/2002 | 11:26 AM | 4,144 | itprev40.DLL |
| 01/23/2002 | 12:00 PM | 4,560 | itprev50.DLL |
| 01/23/2002 | 12:01 PM | 4,864 | itprev55.DLL |
| 01/23/2002 | 12:01 PM | 4,864 | itprev60.DLL |

## Technical Support

### Email

We offer technical support by email. Please email your questions to [support@icetips.com](mailto:support@icetips.com) and we will get back to you as soon as possible. We usually respond to technical support emails within an hour.

### On the web

#### Icetips Software

<http://www.icetips.com>

The Icetips website contains a lot of articles about programming in Clarion and we are beginning to put up articles that only apply to our products. You will be able to find a link on each product page and also in the Product menu.

#### Icetips Bulletin Board

<http://www.icetips.com/bboard/index.php>

This bulletin board has sections for each of our products. It is open for everyone to post and reply to questions.

#### Icetips Software Online Support

<http://www.icetips.com/supportentry.php>

We would appreciate if you consider using these resources as well as our online knowledgebase before posting on the newsgroups. We usually respond to emails within an hour from 9:00 to 19:00 (9am to 7pm) in the US Central time zone (GMT -6:00hr) Messages sent through the I-SOS are logged into a database and forwarded to us via email, but because they are logged into a database we do not have to rely 100% on just the email part of it.

### Newsgroups

You can also post questions on the `softvelocity.clarion.third_party` newsgroup on the `news.softvelocity.com` news server or `comp.lang.clarion`, which you can get to at that same news server or on the web at:

<http://groups.google.com/groups?hl=en&group=comp.lang.clarion>

### Mail

If you feel that you need to send us something in regular mail, please address it to:

Icetips Software  
4915 Kenton Harbor  
San Antonio, TX-78240  
United States of America