THINKDATA INC.

Clarion Third Party Add-Ons

EnhancedScrollClass User Guide

CLARION THIRD PARTY ADD-ONS

EnhancedScrollClass User Guide

2508 Pacific Avenue • Suite 1
Venice Beach, California 90291
Phone 310.823.2571
info@thinkdata.com

Table of Contents

Introduction

Overview

The EnhancedScrollClass has been designed to provide high performance browsing and scrolling on page loaded browses in a Clarion 5.5 application regardless of the file driver being used by the developer. It is a replacement for the standard ABC BrowseClass and its central role is to use Windows timers to determine when an end user actually starts and stops a scroll operation on a list control. This effectively eliminates the unnecessary secondary lookups performed on a browse when a user scrolls through it. Integrating the EnhancedScrollClass into an existing application is easy since it has been derived from the standard BrowseClass.

Installation

To install the EnhancedScrollClass, simply execute the supplied escrollsetup.exe file and follow the instructions in the installation program.

Two changes must be made to the shipping copy of abbrowse.inc to allow for derivation of the base BrowseClass. A sample copy of a modified abbrowse.inc will be placed in the installation directory (i.e. C55\LibSrc\EScroll) for your review. Figure 1.1 shows the first change which must be made to abbrowse.inc for the EnhancedScrollClass to function properly. The second change is shown in Figure 1.2. Please make these changes before using the EnhancedScrollClass for the first time.

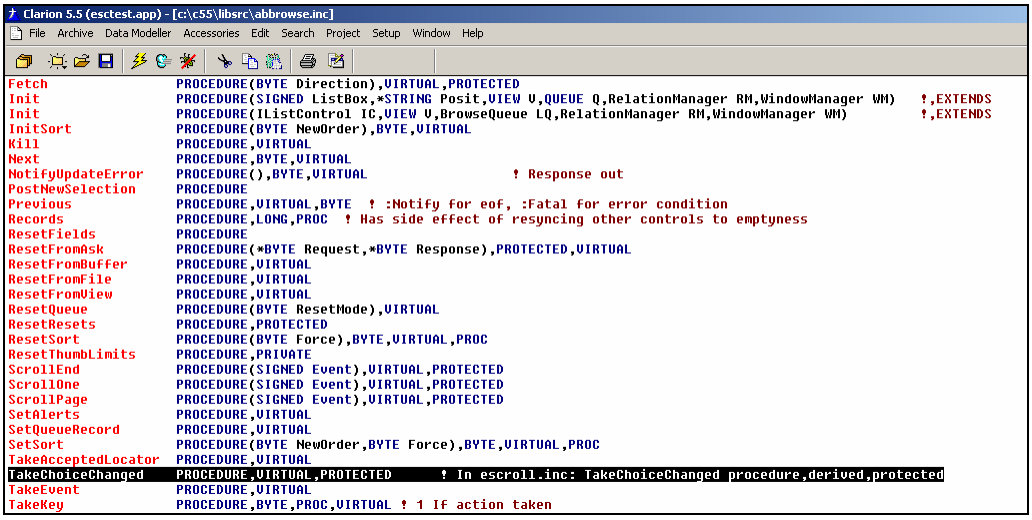

FIGURE 1.1 The TakeChoiceChanged method must be made VIRTUAL and PROTECTED. This will allow the EnhancedScrollClass to derive it.

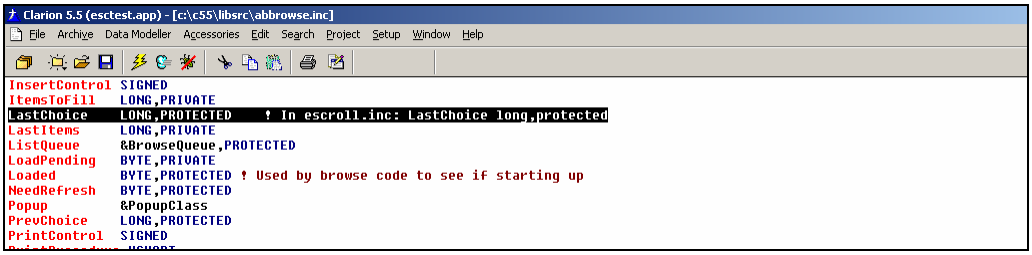

FIGURE 1.2 The LastChoice property must be made PROTECTED to allow derived class methods to access it.

Using the EnhancedScrollClass

The EnhancedScrollClass was designed to be a plug-in type component so a developer can attain maximum benefit with very little effort. Figure 1.3 shows how to convert a standard browse to an EnhancedScrollClass browse. Simply right click on the list box, select Properties and scroll to the Classes tab. Then uncheck the Use Default ABC BrowseClass checkbox and check the Use Application Builder Class checkbox. Then find and select the EnhancedScrollClass in the list.

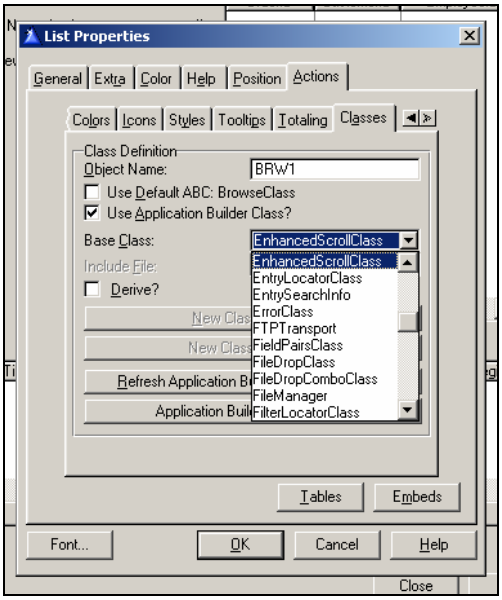

FIGURE 1.3 Converting a standard browse to an EnhancedScrollClass using the Application Builder Class dropdown on the Classes tab in List Properties dialog

A new method has been added to the EnhancedScrollClass called TakeLastNewSelection. It is similar to the standard BrowseClass.TakeNewSelection method except that it is controlled by a Windows API timer and the method is guaranteed to fire when the end user has stopped scrolling the browse. You will experience best performance by moving source code from the TakeNewSelection method to the TakeLastNewSelection method. Figure 1.4 shows this new method in the Embedded Source view.

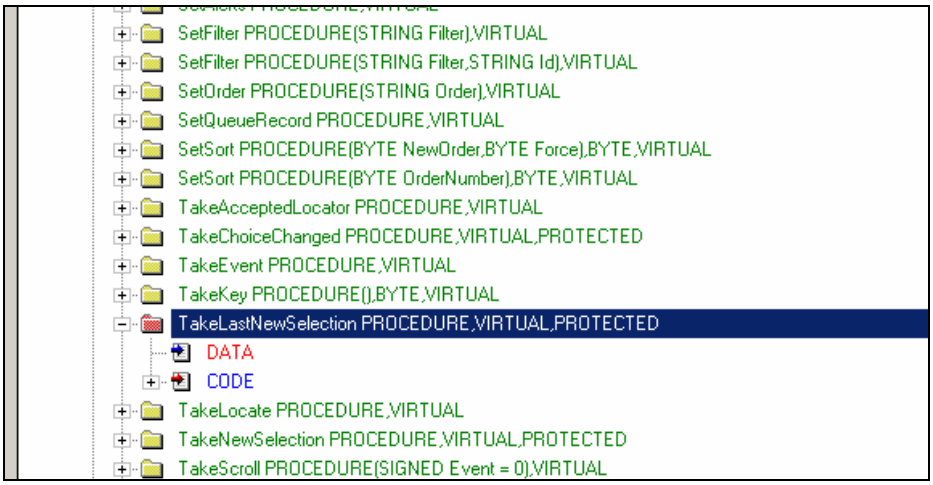

FIGURE 1.4 For best performance the source code from the TakeNewSelection method should be moved to the TakeLastNewSelection method.

Example Application

The EnhancedScrollClass ships with an example application called esctest.app. This application demonstrates the functionality of the EnhancedScrollClass compared with a native BrowseClass on the same window. The performance is most notably improved in SQL based applications. This can be seen in the demo application for all users of Microsoft SQL Server. Simply select the MSSQL Demo (Northwind DB) from the Browse menu after compiling the esctest.app application. Scroll through the standard browse with the up/down arrows on the keyboard or Click the scroll arrows with the mouse. Using the up/down arrow keys you will find that the TakeNewSelection code is called for each and every selection regardless of where the user stops. Try the same thing with the EnhancedScrollClass. The TakeLastNewSelection code is not called until the user discontinues the scroll operation. This improves performance dramatically when using SQL file drivers because the amount of network traffic is greatly reduced.

Overriding Default Behavior

There are two properties in the class which can be overridden in the constructor method which define how quickly a scroll is taken by the EnhancedScrollClass. Figure 1.5 shows an example of an overridden constructor method as seen in the example application esctest.app on the EnhancedScrollClass control. The first value, self.dwMilliseconds, represents the amount of time that needs to elapse with no scrolling for the EnhancedScrollClass.TakeLastNewSelection method to fire. The second value, self.dwTimerMilliseconds, represents the amount of time that needs to elapse to begin a scroll operation process. Experimenting with these values can lead to higher performance depending on the size of the dataset and the number of rows being fetched at any given time.

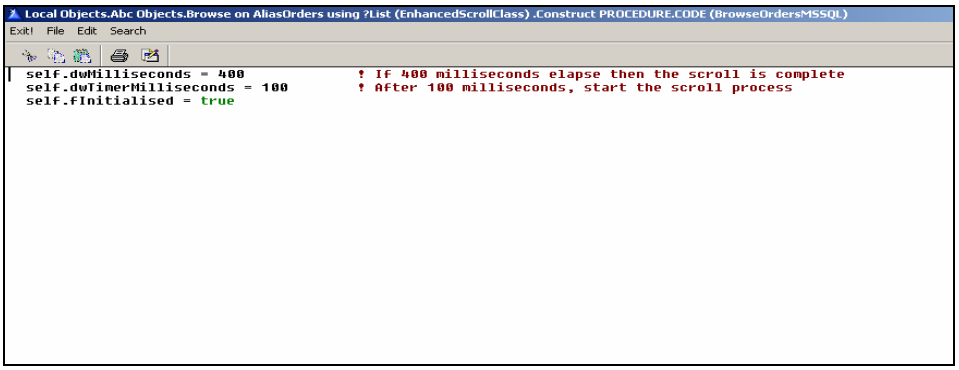

FIGURE 1.5 The code in the EnhancedScrollClass.Construct method overrides the default timer behavior of the class