

XPanel

XPanel for FLEX/Flash
Copyright ©2005-2006 by CTI
Copyright ©2006 by Farata Systems, Inc

Description

XPanel allows to trace Flex/Flash applications interactively, with better performance and control. It provides API familiar to J2EE developers for Flex 1.* users and allows to add Flex 2 compatible end-user tracing/logging to the application using non-debug player.

Package consists of XPanel.exe located in /executable folder and XPanel_Library.swc located in /bin folder.

The other files in distribution represent samples, sources and configurations

Requirements

XPanel runs on Windows 95/98/Me, NT/2000/XP and does not require any installation. It requires .NET Framework 1.1 run-time preinstalled on the machine - computers with XP or Microsoft Office have it.

When started, XPanel will show empty window with toolbar. It now can accept trace messages from your Flex applications.

Usage

The XPanel_Library.swc includes two AS objects : Logger.as and XPanelTarget.as. They can also be included into your projects in the source form – please see /src folder in distribution. Logger can be used with Flex 1.* applications and can be “wrapped” for regular Flash. XPanelTarget is for Flex 2 only at this point

Using Logger Object

Logging class uses simplified API of popular Java logging packages :

It provides 5 static methods *debug(o:*)*, *info(o:*)*, *warn(o:*)*, *error(o:*)* and *fatal(o:*)*

Here is sample program using those APIs

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
  <mx:Button x="65" y="62" label="Debug" click="doDebug()" />
  <mx:Button x="237" y="62" label="Warn" click="doWarn()" />
  <mx:Button x="321" y="62" label="Error" click="doError()" />
  <mx:Button x="410" y="62" label="Fatal" click="doFatal()" />
  <mx:Button x="158" y="62" label="Info" click="doInfo()" />
</mx:Application>
```

```

<mx:Script>
    <![CDATA[
import com.theriabook.utils.Logger;
private function doDebug():void
{
    Logger.trace("debug1");
}

private function doInfo():void
{
    Logger.info("info1");
}

private function doWarn():void
{
    Logger.warn("warn1");
}

private function doError():void
{
    Logger.error("error1");
}

private function doFatal():void
{
    Logger.fatal("fatal1");
}

    ]]>
</mx:Script>
</mx:Application>

```

Using XPanelTarget

XPanelTarget Class is very basic implementation of AbstractTarget class in Flex 2. It communicates logEvents from the application to the XPanel interface.

Here is an example of the usage:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application
    xmlns:mx="http://www.adobe.com/2006/mxml"
    xmlns:xpanel="http://www.faratasystems.com/xpanel"
    horizontalAlign="center"
    verticalAlign="middle"

```

```

layout="horizontal">
<mx:Script>
    <![CDATA[
        import mx.logging.Log;
        import mx.logging.ILogger;
        import mx.logging.LogEventLevel;

        private static const aLogger:ILogger =
            Log.getLogger ("testXPanelTarget");

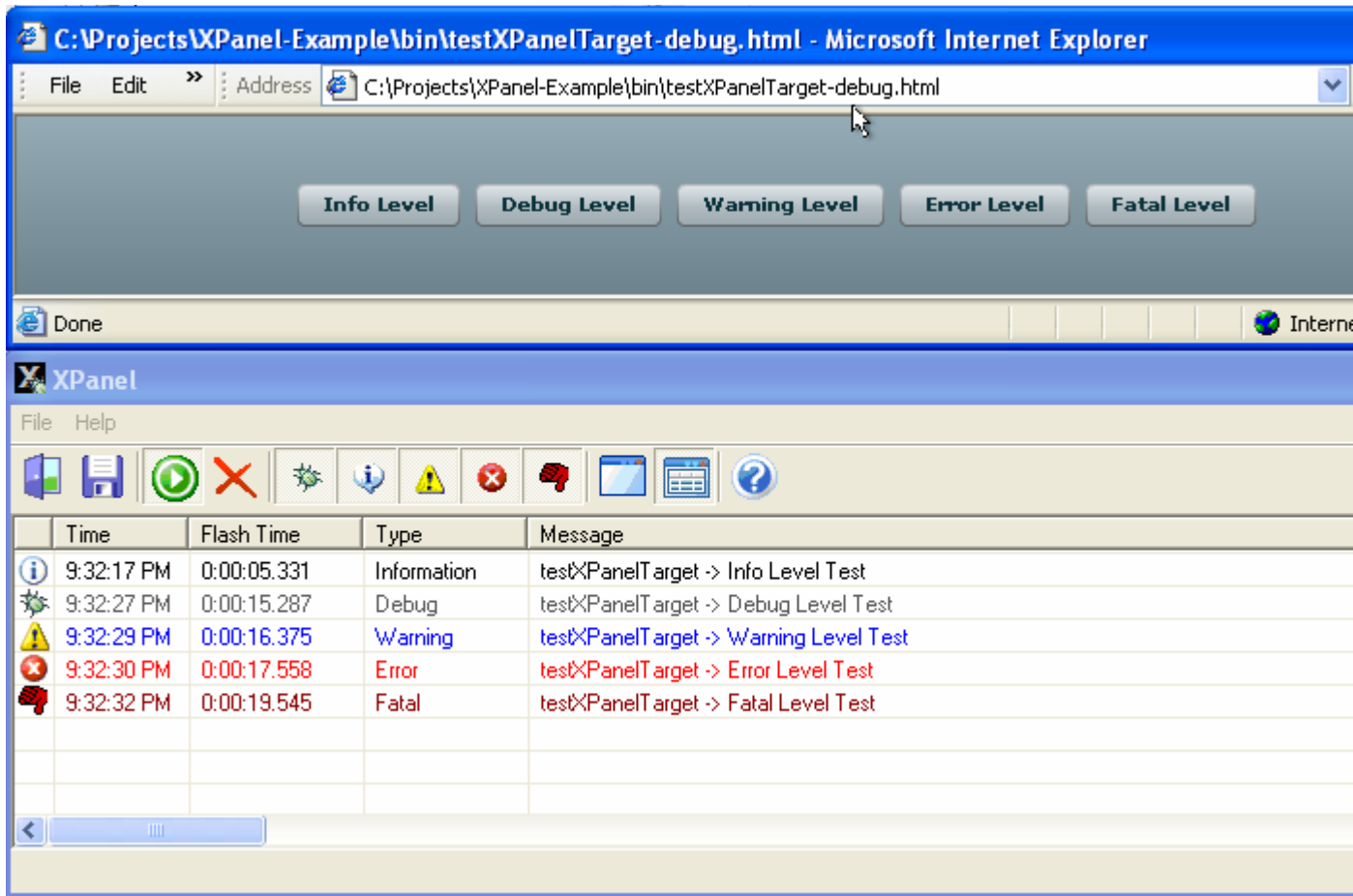
    ]]>
</mx:Script>
<xpanel:XPanelTarget
    fieldSeparator=" -> "
    includeCategory="true"
    level="{LogEventLevel.DEBUG}">
    <xpanel:filters>
        <mx:Array>
            <mx:String>*</mx:String>
        </mx:Array>
    </xpanel:filters>
</xpanel:XPanelTarget>
<mx:Button
    label="Info Level"
    click="aLogger.info ('Info Level Test')"/>
<mx:Button
    label="Debug Level"
    click="aLogger.debug ('Debug Level Test')"/>
<mx:Button
    label="Warning Level"
    click="aLogger.warn ('Warning Level Test')"/>
<mx:Button
    label="Error Level"
    click="aLogger.error ('Error Level Test')"/>
<mx:Button
    label="Fatal Level"
    click="aLogger.fatal ('Fatal Level Test')"/>
</mx:Application>

```

Testing

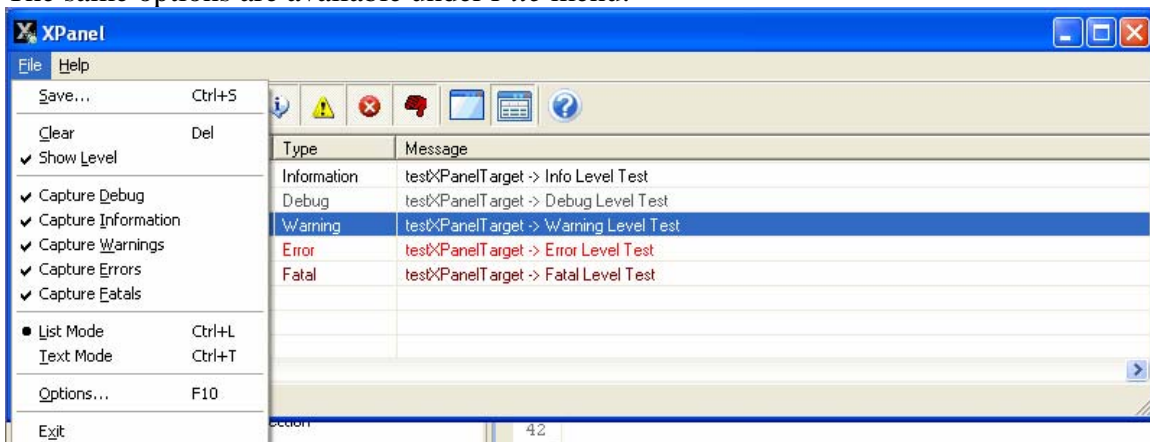
Start XPanel.exe and either of the above samples (sources are located in /examples, compiled SWF versions are in /examples/bin)

Running either of the samples above should see two windows:

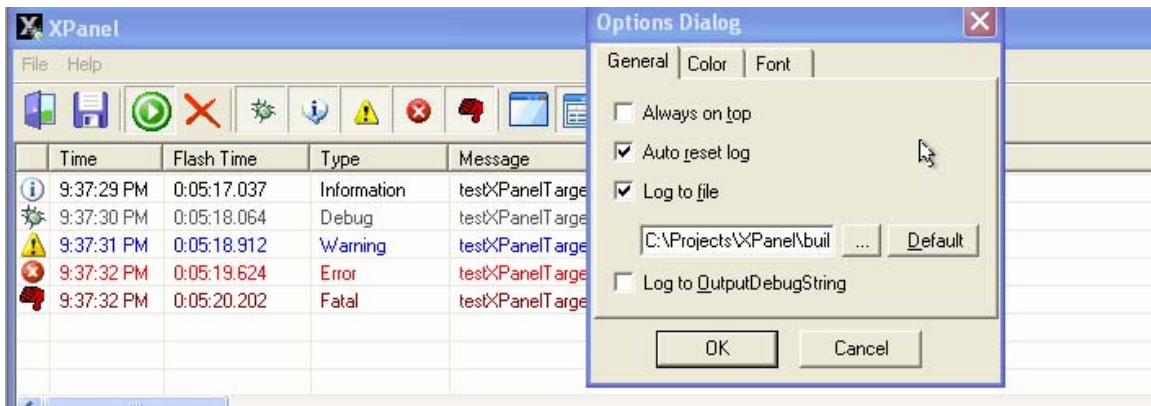


Toolbar provides quick access to *Exit*, *Save Log*, *Start/Stop Logging*, *Clear Log*, *independent logging levels*, *Text/DataGrid Mode* and “*About*” menu options

The same options are available under *File* menu:



Finally, common running options are available under *File->Options (F10)* menu and include options to automatically write the log to disk/use OS debug console and customize the UI:



Enjoy

LIABILITY DISCLAIMER -- READ BEFORE INSTALLING XPanel

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, INCLUDING AND WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL THE AUTHOR OR HIS COMPANY BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, LOSS OF INFORMATION, OR ANY OTHER LOSS) , WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR INABILITY TO USE THIS SOFTWARE.

Copyright © 1996-2006 Computer Technology, Inc.

Copyright © 2006 Farata Systems, Inc. All rights reserved.