



AltioLive Developer Training

Version 5.4

9. Debugging your Application

Summary

Complex applications can be built using the AltioLive Studio. During the development of an application it is likely that errors occur. This session shows us where to look for debug and logging information that can be a valuable aid to problem solving.

Integra SP

88 Wood Street London
EC2V 7RS
United Kingdom

www.altio.com

tel: +44 (0) 20 8528 1045

Contents

The Designer in Debug Mode	2
Viewing Client Data	2
Debug Messages at the Java Console	3
Practical Exercise: Trapping errors at the Java Console.....	4
The Client in Debug Mode	5
Application Logging.....	6
Practical Exercise: Explore Application Logging	7
Presentation Server Logging	8
Log File Management.....	9
Troubleshooting.....	10
Can't run Designer?.....	10
Getting a recurring problem that you think you've fixed?	10
Cannot see your data?	10
Browser Problem?.....	10
Contact Altio Support.....	10
Altio Developer Forum	11

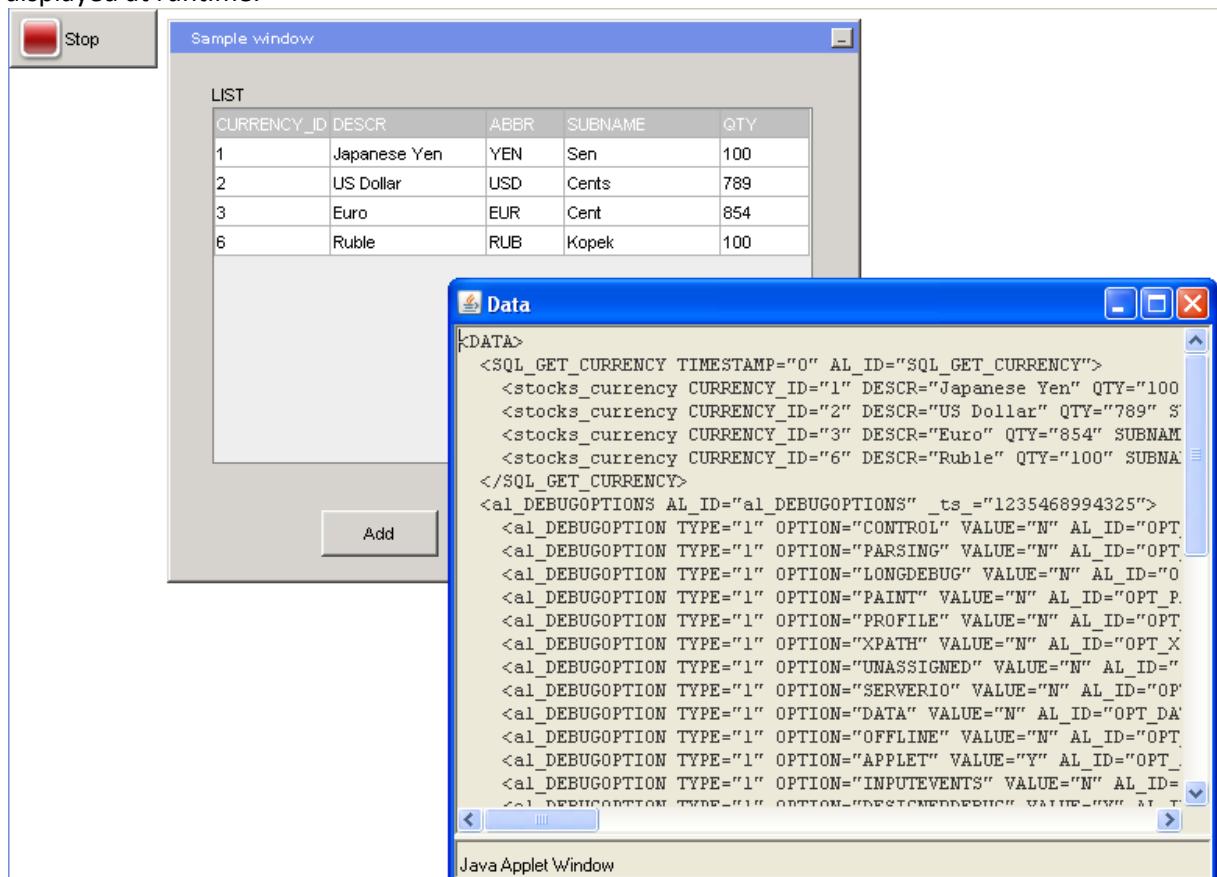
The Designer in Debug Mode

Viewing Client Data

In the Designer, the client data and configuration are both available from the **Debug** menu.

The debug window is also available in Run mode.

1. Press **Ctrl+Shift+D** to open a window that displays the current data view at the client. The window provides a useful check, for example, if data is not displaying correctly in a control. Seeing if data is present for an application view will help determine if a problem lies with a service request or the data properties for a control. Here's an example of a **Data** window displayed at runtime:



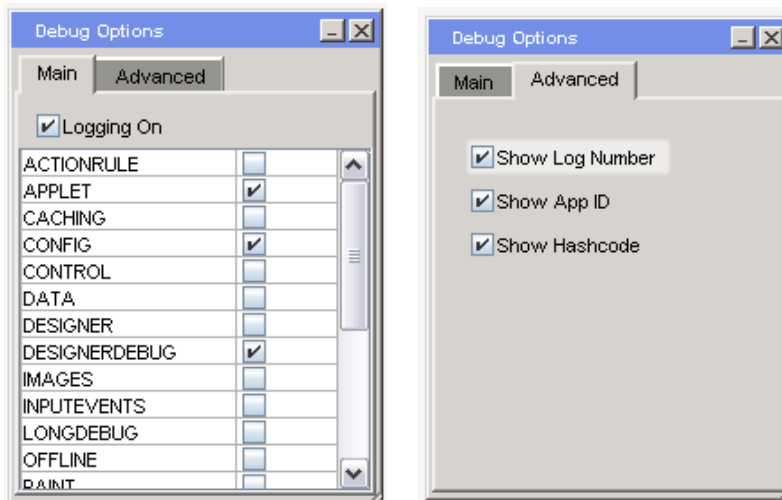
Debug Messages at the Java Console

Once the Designer is running in debug mode then the Java Console can be set to show debug messages. It is possible to view many types of messages at the console, however it's unlikely that we would view all messages types together as this can result in a tremendous volume of output displayed.

There is a debug setting window that can be accessed in the Designer by:

- Clicking on the **Debug | Debug Settings** menu item.
- At run-time by pressing **ctrl+shift+L**.

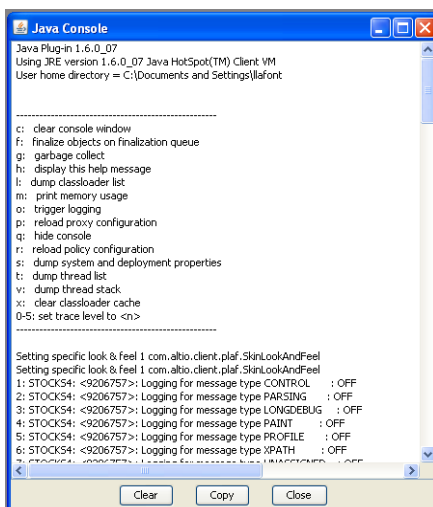
The Debug window:



To View messages at the Java Console:

1. Load an application into the Designer.
2. Display the **Java Console**. From the **Tools** menu of your internet browser, select:
 - **Sun Java Console** for Internet Explorer browsers,
 - **Java Console** for Firefox browsers.

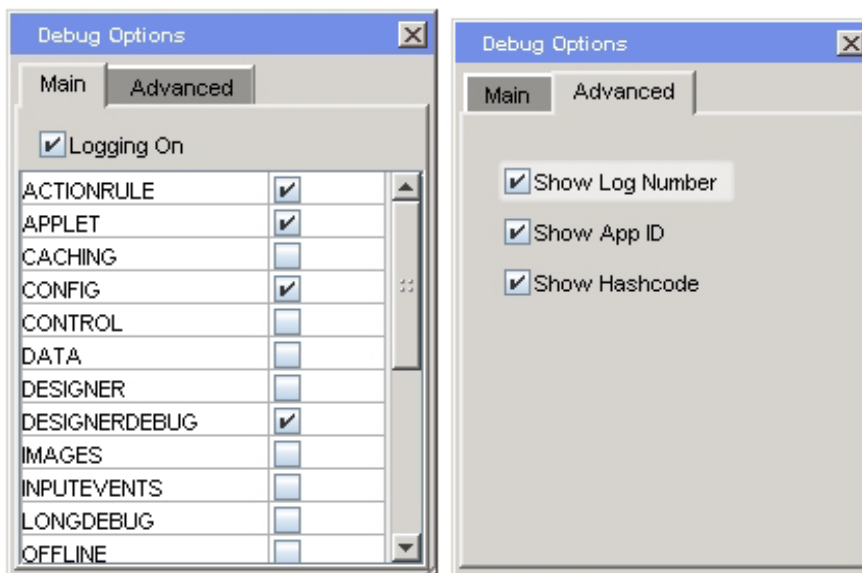
The Java console should display:



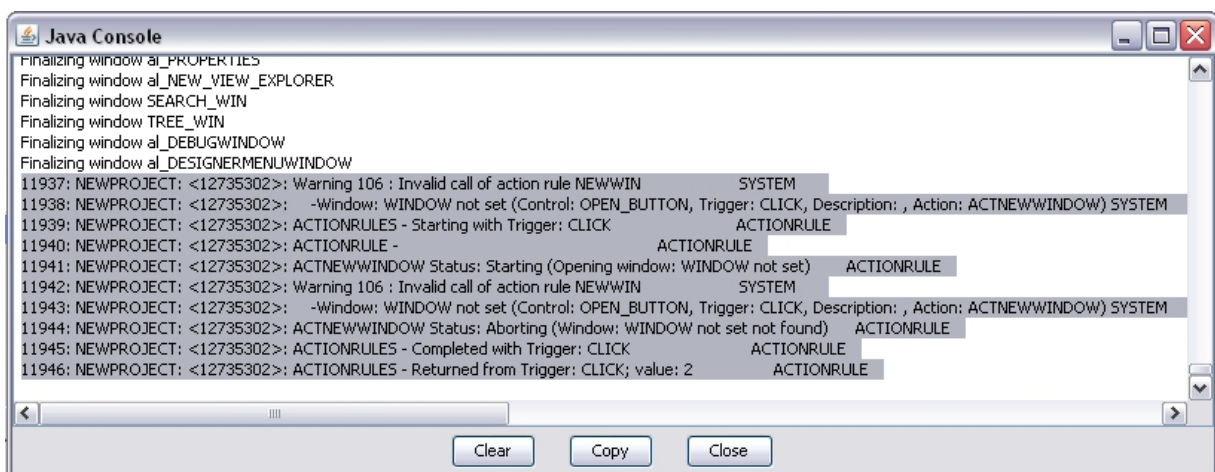
Practical Exercise: Trapping errors at the Java Console

To investigate messages at the Java Console, we will filter the Java Console output to include ACTIONRULE messages. We will purposely create an Action that has an error and observe the output at the Console.

1. From the AltioLive Studio, create a new project.
2. Open this new project with the Designer.
3. From the **View Explorer** window, expand the **Windows** folder and double-click on the default window.
4. Add a button control to the default window.
5. Display the properties of the button, and go to the **Events** tab.
6. On the **Click** event add an **Open window** action and do not set any parameter for this action.
7. Run the view.
8. While the application is running, press **Ctrl-Shift-L** to bring up the **Debug Options** window.
9. Set its parameter as shown:



10. Click on the button you have created.
11. Display the Java console, it should display error messages like this:



The Client in Debug Mode

The AltioLive Client may be run in Debug mode; this can be invaluable for tracking problems that have escaped application testing.

To use the Debug mode:

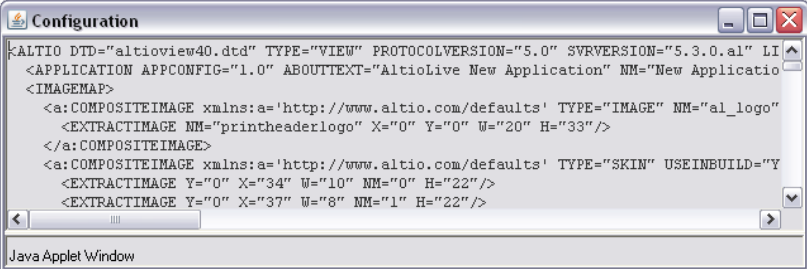
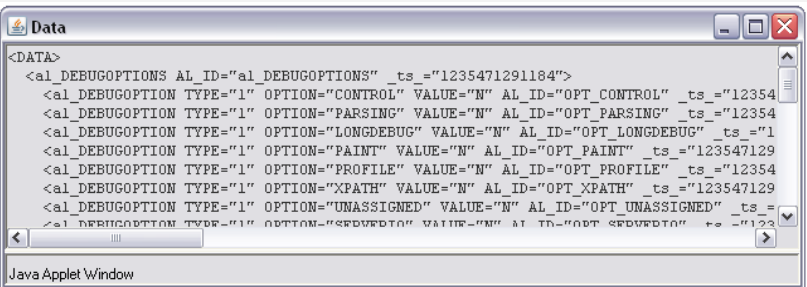

1. Go to your installation folder and open the **htmltemplate.txt** file. Each application has an **htmltemplate.txt** file located at its 'root' directory (for example: **Altio54\WEB-INF\classes\apps\myApplication\htmltemplate.txt**).
2. Set the applet attribute called **useReleaseJars** to **false**:

```
var appletWidth = screen.width;
var appletHeight = screen.height;

var useReleaseJars = false;           // Set to false to use debug jars
var useSignedJars = false;           // Set to true to use signed jars
var useDesignerJars = false;         // Set to true for the designer
var overrideJarName = "";            // Put in a jar name here only if it is not one of the default jars
var debugVariables = "!ALL BASIC";   // Possible values are shown below
```

3. Restart your application.

In addition, the following key combinations will work in debug mode:

<p>Ctrl-Shift-C</p>	<p>Show client configuration (view xml)</p>	
<p>Ctrl-Shift-D</p>	<p>Show client data</p>	
<p>Ctrl-Shift-L</p>	<p>Change logging options</p>	

Please note: There are performance implications when using the debug jar so its use is not recommended once an application is deployed.

Application Logging

Additional logging relating to the application, service requests and datapools are available for each AltioLive application.

Log files for each application are located at the Applications directory **/appname/log** sub-directory. New log files are created each day.

The logging detail can be adjusted with the Application Manager:

1. Select one of your projects and launch the Application Manager.
2. From the **Application Explorer**, expand the **Logging** folder.
3. Select a Log file and display its properties.

From the **Level** drop-down menu you can choose from the following levels:

NONE	No logging
ERROR	Error messages are logged
WARNING	Error messages and warnings are logged
INFO	Errors, warnings and general messages are logged
DEBUG	Errors, warnings, general and debugging messages are logged

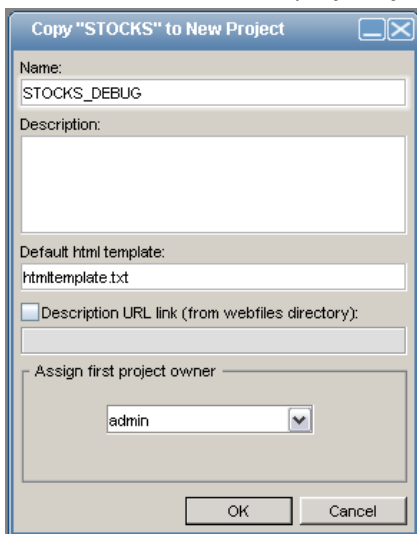
By default, all logging is routed to the one file for the application. If required, specific areas may be logged to separate files by setting the file names in the Application Manager.

Practical Exercise: Explore Application Logging


This practical exercise simulates an error with Live Data updates. In the Practical we will force an error with a connection for a Datapool, and then investigate where the error is displayed in the AltioLive environment.

The practical assumes that you have completed the session on Datapools and have the Stocks application working with Live updates.


Please note: You may want to copy your project for this exercise: to do so, from AltioLive Studio, select the **STOCKS** application, right-click on it and select **Copy to New Project** in the context menu. Give a new name to this project, for example **STOCKS_DEBUG**.



To 'break' the datapool:

1. Open the **Stocks_Debug** application with the **Application Manager**
2. Open the **DP_GET_STOCKS** service function.
3. Add a couple of random characters to the end of the URL.
4. Close the Service function window.
5. Validate the file by clicking on .
6. Save and update the application configuration.
7. Terminate any application instances that might be active.
8. Logout.

We can now test the application. You should see that data is present but Live Updates are not working. We will now investigate the messages available:

1. From AltioLive Studio, expand the **Stocks** node.
2. Double click on the **view.xml** file to launch the **Designer**.
3. Run the view .
4. Check the Data by pressing **Ctrl + Shift + D**. Observe that the initial data has been loaded.
5. Check the Java Console. Are there any Datapool-specific messages displayed?

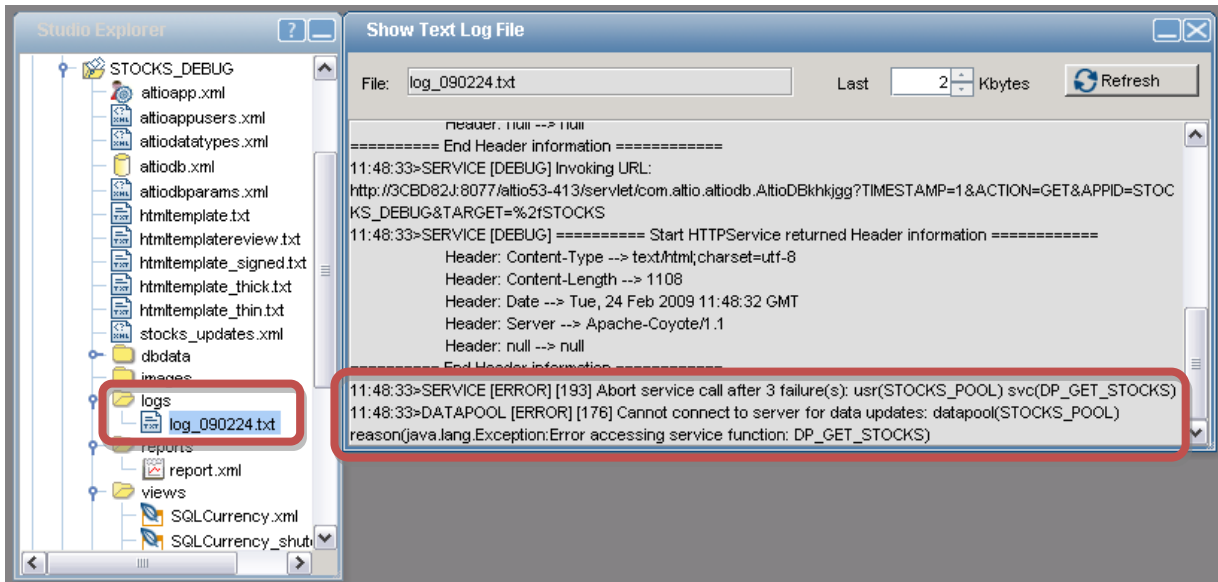
6. Leave the designer.

We have checked the Java Console and Data window but there is no indication of why the live data is failing. Now we will check the application log file:

7. From the **AltioLive Studio**, expand your project node.

8. Expand the **logs** node and open the most recent log file.

You should see an error reported from the Datapool:



There will most likely be multiple errors for each poll of the Back end application by the datapool.

Please note: If you have not created a copy of your **Stocks** application, remember to go back to the Application Manager and correct the URL!

Presentation Server Logging

System administration logging is configured at the Presentation Server Administration Tool. It is unlikely that we would look to this area of logging while developing and testing; however, here is a brief overview.

There are six areas logged:

- **PROFILE:** This log records Presentation Server actions; including the time taken to load application data, and the timing of service requests
- **LICENSE:** This is for product licensing information
- **LOGIN:** This tracks who has logged onto the Presentation Server
- **ALTIODB:** Logs transactions from the AltioDB (if used)
- **DESIGNER:** Logs messages generated by the Designer and
- **MAINLOGGER:** All the logs from the five areas are mirrored in the MAINLOGGER, which is the principal log file for the Presentation Server.

Log File Management

Log files consume server space; each day new sets of log files are created. Remember to reset the logging levels to Error after debugging is completed.

Troubleshooting

Familiarity with the **AltioLive Studio** and the information provided via the **Java Console** and logs will help solve the majority of 'bugs' that occur in an application. Sometimes a problem lies outside this domain; here are a few guidelines from the Altio Support Team:

Can't run Designer?

- Have you been editing your XML files by hand? You may have made a typing error that means the Designer cannot read the file. Make a backup copy of all files before you edit by hand. To check if XML is well formed, open it in a browser such as Microsoft Internet Explorer. If you can see all the XML then it is well formed.
- Is the web server and servlet engine running? Does the Servlet engine need restarting?

Getting a recurring problem that you think you've fixed?

- Have you restarted the servlet engine?
- Delete all your temporary Internet files, as the page may be cached.

Cannot see your data?

- Test your service function and datapool configurations in the **Application Manager**.
- If you are using the Prototyping kit, confirm that the data files are referenced in the **AltioDB** Table Data window (underlying file: altiodb.xml).
- Confirm that the Service Functions are included in the **Initial Data Service** node of the Designer (they will be included in the file `<app_name>_data.xml` that accompanies the view configuration file in the application views directory). For data updates confirm the view subscribes to the datapool.
- Check the controls correctly reference the data (problems can result when the data structure has been changed).
- Check the contents of your data files to confirm that the data is present and well formed. A Service Function to add, update or delete data can easily affect the entire structure of the data if it is written incorrectly.
- Make sure you have **Datakeys** properly defined for all elements.

Browser Problem?

The majority of version 4 and later browsers are able to host the client. If you are not sure that your browser is able to run Altio applications, try accessing the demonstrations at www.altio.com. While AltioLive should run on any browser/jvm (1.1 or later), there is an infinite combination of possibilities. If you continue to have problems, please contact support@altio.com.

Contact Altio Support

If you still have a problem then email Altio support at support@altio.com including:

- The platform you are working on, the web browser and servlet engine you are using.
- The version of AltioLive you are running.
- A clear description of the problem and how to replicate it.
- Relevant log files, java console output and view configuration files.

Altio Developer Forum

Note that we now have a developer forum at <http://www.altio.com/forums/>. This is a useful source of tips and comments from users and responses from Altio developers and Altio support staff.

Next Step: Further Exercises

AltioLive Developer Training

Document Information

KEYWORDS: Debug, Java Console, Logging, Troubleshooting.

Integra SP – Altio

Telephone: +44 (0) 20 8528 1045

Internet: www.altio.com

Copyright © 2010 Integra SP

Copyright in this document is vested in Integra SP. The contents of the document (wholly or in part) must not be reproduced, distributed, used or disclosed without the prior written permission of Integra SP.

Integra recognizes the trademarks or registered trademarks of any third party product or company name referenced in this document at the time of its publication.