

Using "DebugView" to review event calls in Orixo Apps and track down difficult bugs

Sometimes Orixo Apps will fail with error messages which are vague or unclear. In these cases it can be hard to locate the reason for the error.

In such cases it can be useful to use a program called "DebugView", which is provided freely by Microsoft, to review events that are called by your program before and after the error. The list of event calls can be saved as a text file and sent to Orixo developers for review. This should enable your Orixo App to be fixed. Sometimes as soon as you see the event call log yourself, the reason for the error may become obvious and allow you to fix the issue without further input from Orixo Developers.

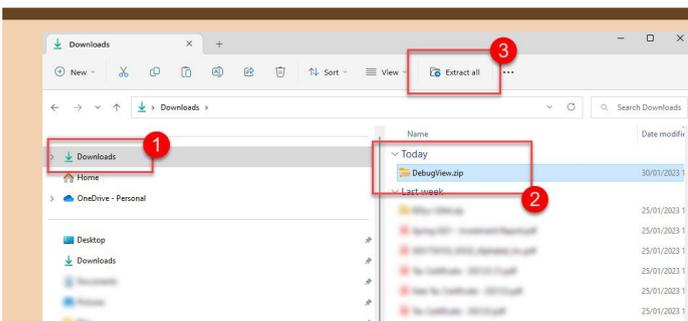
The DebugView program

This is a freeware program distributed by Microsoft. It is a very small exe file which is part of their "SysInternals" developers toolset. You can read more about it here: [SysInternals - Debugview](#)

The Debugview.exe file is included in a zip file here: [DebugView.zip](#)

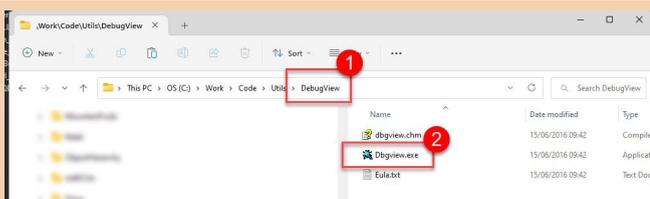
Or you can use the link to "SysInternals" above to run the same type of debugging and testing procedure using an on-line tool.

Downloading and using DebugView.exe



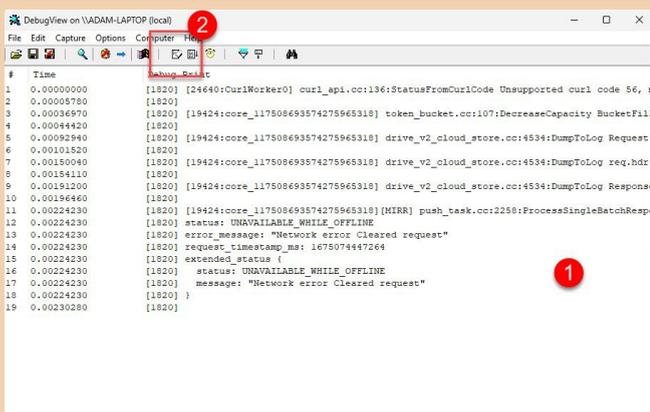
Downloaded DebugView.exe

1. The Zip file will be downloaded to a location on the user's computer.
2. Click on the ZIP file.
3. Select "Extract All" from the menu, or right click on the ZIP file and select "Extract all"



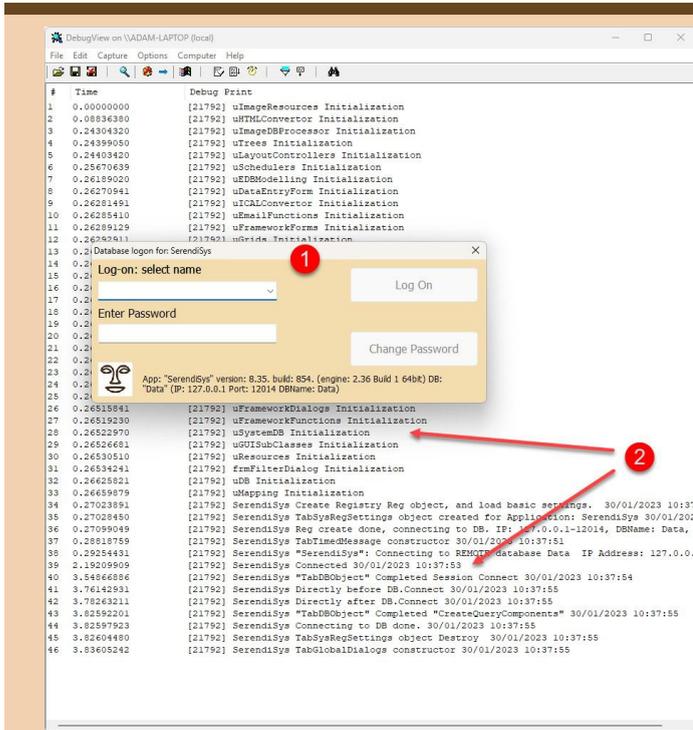
Running the DebugView.exe

1. Items in the ZIP file will be extracted into a folder chosen by the user.
2. The main executable file. Double-click on this to run the program.



Debug View Program running

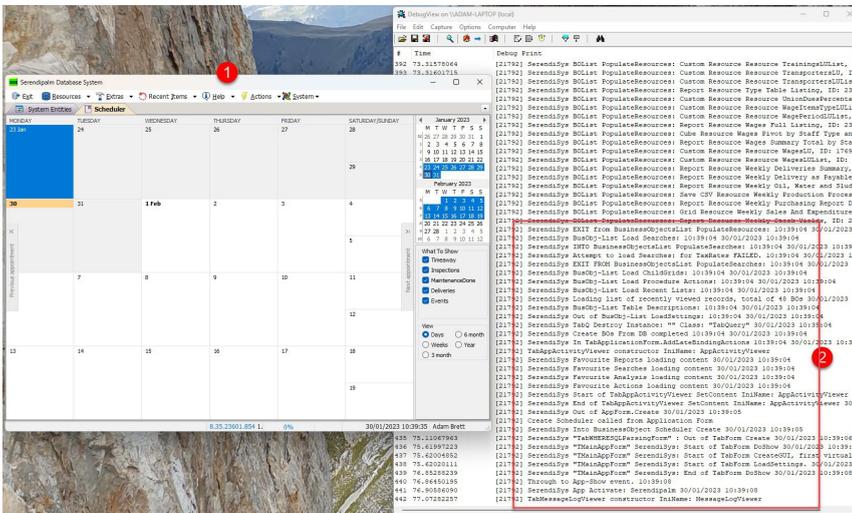
1. The main window of the debug view program.
2. The "Clear All" button in the toolbar. Note that this button can be used to clear the event call log if a large number of messages build up.



Messages showing in the Debug View Window

1. The user has run their Orixia App.
 2. The Event Call Log is filling with messages output from the App.
- Note that these messages are a normal part of the operation of the App. When there is a hard-to-fix issue with an App, it is often useful for Orixia Developers to see the output of this screen.

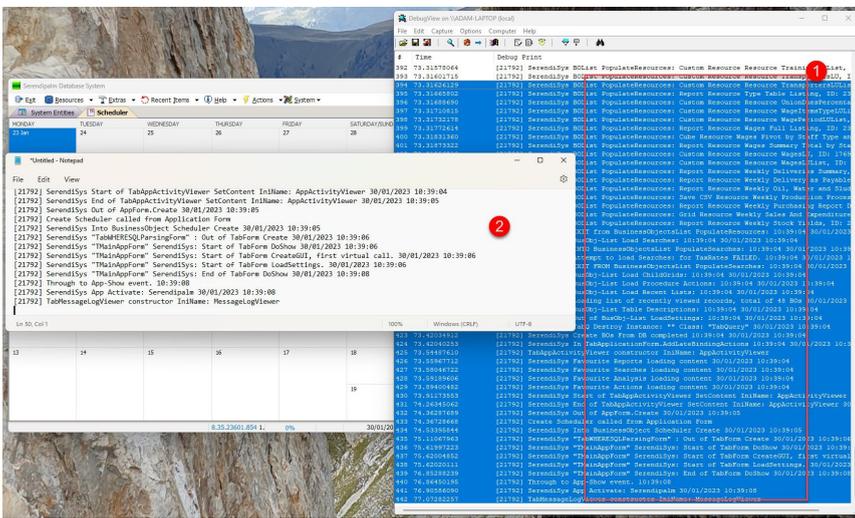
Reviewing the Event Call Log



Debug View Event Call Log

1. A running Orixia App with the "Scheduler" showing.
2. The Event Call Log, with multiple messages.

As an Orixia App loads, its source code contains calls which can be seen by the DebugView program. DebugView will load all these messages into a Call Log Window. When there is an issue with an App it may be because some of these events are mis-firing. To give the Orixia Developer a chance to understand the issue, **save** the messages in the Event Call Log to a file on-disk which can be sent to your developer to review. If there are a fairly small number of messages send the whole file.



Paste Lines from the Event Call Log into Notepad

If the error only occurs after a large number of steps has been undertaken the event call log may contain a very large number of messages. In this case it is usually sufficient to copy the last 30 - 50 lines of messages into Notepad and send just these.

1. Click on the last line in the Event Call Log. Hold down the Shift key and press the up arrow to selected lines of text.
2. Copy these to the clip-board and paste them into any suitable program such as Notepad or Word. Then email this file to your Developer.

Understanding the contents of the Event Call Log

Example Lines from the Call Log: Database Connection

```
[23308] uResources Initialization
[23308] frmFilterDialog Initialization
[23308] uDB Initialization
[23308] uMapping Initialization
[23308] SerendiSys Create Registry Reg object, and load basic settings. 30/01/2023 11:11:28
[23308] SerendiSys TabSysRegSettings object created for Application: SerendiSys 30/01/2023 11:11:28
[23308] SerendiSys Reg create done, connecting to DB. IP: 127.0.0.1-12014, DBName: Data, Signature: orixa-server 30/01/2023 11:11:28
[23308] SerendiSys TabTimedMessage constructor 30/01/2023 11:11:28
[23308] SerendiSys "SerendiSys": Connecting to REMOTE database Data IP Address: 127.0.0.1 Port: 12014 30/01/2023 11:11:28
[23308] SerendiSys Connected 30/01/2023 11:11:30
[23308] SerendiSys "TabDBObject" Completed Session Connect 30/01/2023 11:11:31
[23308] SerendiSys Directly before DB.Connect 30/01/2023 11:11:31
[23308] SerendiSys Directly after DB.Connect 30/01/2023 11:11:31
[23308] SerendiSys "TabDBObject" Completed "CreateQueryComponents" 30/01/2023 11:11:31
[23308] SerendiSys Connecting to DB done. 30/01/2023 11:11:31
[23308] SerendiSys TabSysRegSettings object Destroy 30/01/2023 11:11:31
[23308] SerendiSys TabGlobalDialogs constructor 30/01/2023 11:11:31
```

The above lines from a DebugView call log do not contain any errors.

Errors which can occur at this step

When an Orixa App starts it accesses values from the Registry, including a subscription key. If these values are incorrect or missing the App will fail. To fix this, run the "SysMgt.exe" and use it to save the required data to the Registry. In such a case errors would be seen after the line "SerendiSys TabSysRegSettings object created"

When an Orixa App connects to the database errors can also occur. In this case lines would be present to indicate this after the line containing "Conncting to REMOTE database"

Example Lines from the Call Log: Opening User Interface elements in the App

```
[23308] SerendiSys App Activate: Serendipalm 30/01/2023 11:12:07
```

```
[23308] SerendiSys "TPeopleFF" PeopleEF: Start of TabForm DoShow 30/01/2023 11:12:12
[23308] SerendiSys "TPeopleFF" PeopleEF: Start of TabForm CreateGUI, first virtual call. 30/01/2023
11:12:12
[23308] SerendiSys "TPeopleFF" PeopleEF: Start of BusObjForm CreateGUI 30/01/2023 11:12:12
[23308] TabDataviewer constructor IniName:
[23308] SerendiSys "TPeopleFF" PeopleEF: End of BusObjForm CreateGUI 30/01/2023 11:12:12
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutPeople: Start of TabBaseDBLayout.CreateGUI
30/01/2023 11:12:12
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutPeople: Start of
TabBaseDBLayout.CreateDefaultLayoutItems, 0 controls 30/01/2023 11:12:12
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutPeople: End of
TabBaseDBLayout.CreateDefaultLayoutItems, 12 controls 30/01/2023 11:12:13
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutPeople: Start of
TabBusObjDBLayout.CreateDefaultLayoutItems, 12 controls 30/01/2023 11:12:13
[23308] TabTypesQ Global Types list class Create completed, 708 records in the list
[23308] SerendiSys TabQ Destroy Instance: "" Class: "TabSensitiveQuery" 30/01/2023 11:12:13
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutPeople: End of
TabBusObjDBLayout.CreateDefaultLayoutItems, 12 controls 30/01/2023 11:12:13
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutDataLayoutFarmers: Start of
TabBaseDBLayout.CreateGUI 30/01/2023 11:12:13
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutFarmers: Start of
TabBaseDBLayout.CreateDefaultLayoutItems, 0 controls 30/01/2023 11:12:13
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutFarmers: End of
TabBaseDBLayout.CreateDefaultLayoutItems, 16 controls 30/01/2023 11:12:13
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutFarmers: Start of
TabBusObjDBLayout.CreateDefaultLayoutItems, 16 controls 30/01/2023 11:12:13
[23308] TabStatusQ Global Status list class Create completed, 58 records in the list
[23308] SerendiSys TabQ Destroy Instance: "" Class: "TabSensitiveQuery" 30/01/2023 11:12:13
[23308] SerendiSys "TabBusObjDBLayout" DataLayoutFarmers: End of
TabBusObjDBLayout.CreateDefaultLayoutItems, 16 controls 30/01/2023 11:12:13
```

The above lines occur after the "App Activate" message. This message indicates that the whole of the App start up process is complete, and has occurred without errors.

Lines after this line occur as the user opens and uses parts of the App user interface. In the example above the user is opening a **People** edit form. The above code does not contain errors.

Errors which can occur at this step

As user interface elements are created and shown complex calls to source code, database resources and external libraries can occur. Issues with any of these processes will result in error messages which will be visible in the event call log.

If an error message is vague, it is hard to know what has triggered it, but when the call log is viewed the events directly before the error can be seen and this often enables the User or Developer to figure out the cause of the problem.