

Developer Console Basics

Wednesday, April 19, 2017 7:44 AM

Get Started with Dev. Console

Set up a workspace using [Workspace Manager](#). In each workspace, you can create Apex classes, Lightning components, Vforce pages and more.

It's recommended to assign each project their own workspace.

Navigate and Edit Source Code

If the code syntax is incorrect, the Problems tab will show what and where the error is.

Generate and Analyze Logs

Execution Log

Timestamp	Event	Details
17:49:20:053	HEAP_ALLOCATE	[1] Bytes:4
17:49:20:053	HEAP_ALLOCATE	[50] Bytes:5
17:49:20:053	HEAP_ALLOCATE	[56] Bytes:5
17:49:20:053	HEAP_ALLOCATE	[64] Bytes:7
17:49:20:053	SYSTEM_MODE_...	false
17:49:20:053	HEAP_ALLOCATE	[1] Bytes:5
17:49:20:053	CONSTRUCTOR...	[1] 01p50000001H3vF <init>()
17:49:20:053	VARIABLE_SCO...	[1] this EmailMissionSpecialist true false
17:49:20:054	VARIABLE_ASSI...	[1] this {} 0x722ad411
17:49:20:054	HEAP_ALLOCATE	[EXTERNAL] Bytes:6
17:49:20:054	STATEMENT_EX...	[1]
17:49:20:054	CONSTRUCTOR...	[1] 01p50000001H3vF <init>()

☐ This Frame ☐ Executable ☐ Debug Only ☐ Filter [Click here to filter the log](#)

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
Your Name	Unknown	/services/da...	7/22/2016, 5...	Success		7.38 KB

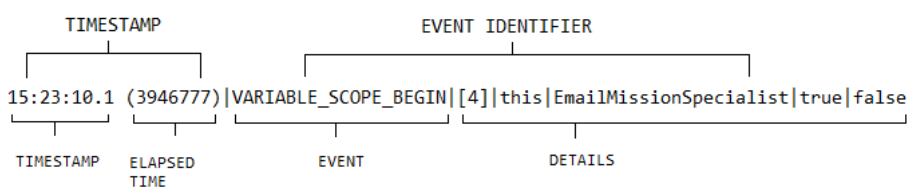
Analyzing logs:

Timestamp - when the error occurred

Event - the action that caused the error

Details - Line of code and the method name where the code was executed.

The image above shows how you can filter the debug log. Filter allows for custom filter.
File -> Open Raw Log to show more information.



Using the Log Inspector (The name of the game)

Debug -> View Log Panels

The screenshot shows the Salesforce Developer Console interface. The top bar displays the file name 'EmailMissionSpecialist.apxc' and the log name 'Log executeAnonymous @7/25/2016, 6:50:29 PM'. The main area is divided into several panels:

- Stack Tree (1):** A tree view showing the hierarchy of executions. It includes columns for Unit, Duration, and Heap. The selected item is 'se...' with a duration of 3.52 and a heap size of 183.
- Execution Stack (2):** A bottom-up view of the selected item. It shows the sequence of calls leading to the current execution. The selected item is 'sendMail' with a duration of 3.52 and a heap size of 183.
- Execution Log (3):** A table of log events. It includes columns for Timestamp, Event, and Details. The selected event is 'METHOD_ENTRY' at 18:50:29:004.
- Source (4):** The source code file being executed. It shows the line of code being run when the selected event was generated. The selected line is 'em.sendMail('testingemail', 'Fl...'
- Source List (5):** A list of source files. It includes columns for Name and Display Name. The selected item is 'execute_anonym...' with a display name of 'execute_anonym...'.
- Variables (6):** A table of variables and their values. It includes columns for Variable and Value. The selected variable is '<empty>' with a value of 'Must trace APEX_CODE...'.
- Execution Overview (7):** A table showing stats for code being executed. It includes columns for Subject, Limits, Timeline, and Executed Units. The selected item is 'No save order events'.

1. **Stack Tree** - Shows the hierarchy of executions from each class involved. If one class calls a second class, that second class is shown as a subdirectory of the hierarchy.
2. **Execution Stack** - Displays a bottom-up view of the selected item in the debug log starting with the lowest level call, followed by the operation that triggered that call.
3. **Execution Log**
4. **Source** - Contents of the source log file. Indicates the line of code being run when the selected entry was generated.
5. **Source List** - Shows the context of the code being executed when the event was logged.
6. **Variables** - Shows the variables and the values assigned to them.
7. **Execution Overview** - Shows stats for code being executed included heap size and execution time.

Perspective Manager - You can select your preferred perspective that only shows a few of these windows at a time.

Debug -> Save Perspective As - to save my own perspective

Debug -> Switch Perspectives.

Log Levels

Log levels control how much detail is logged for each log category. The following levels are available in the Developer Console, from the least amount of data logged (level = NONE) to the most (level = FINEST).

NONE
ERROR
WARN
INFO
DEBUG
FINE
FINER
FINEST

Logging levels are cumulative. For instance, if the log level is INFO for an event, log information at the ERROR and WARN levels is also included. But if your log level is ERROR, you get only error messages. You don't get warning messages or any other log information for that log category.

The information a log level provides also depends on the log event. Different log events start logging at particular log levels. For instance, some ApexCode events start logging at INFO. If you set ERROR as the log level, you don't get any log information for those events.

To get the information you're looking for, modify the log levels for different events. You want to suppress logging when the robot saves messages about the supposed uprising to the database. So, set the log level for the Database (DB) category to NONE or ERROR.

You can set these levels by selecting **Debug | Change Log Levels**.

Select Log Levels

On the General Trace Settings for You tab, click Add/Change (1).

In the Change Debug Level window, choose the log level for each category. Remember, use log levels judiciously. If your log level is FINEST (2), your code can hit log limits and take longer to run.

Don't worry if you don't see all the levels when you update the log level for a category. Only the levels that add more logging for the category are listed.

Inspect Objects at Checkpoints

You can set breakpoints in your Apex code (limit at 5), however, this option isn't available for VisualForce markup. You can only set breakpoints when your ApexCode log level is set to FINEST.

Heap

Types - Shows how many objects were instantiated and the memory they consumed in **bytes**.

Instances - Shows all instances of this object type.

State - View Object's fields and their values.

The screenshot shows the Salesforce IDE interface with the 'Heap' tab selected. The top bar displays the file 'EmailMissionSpecialist.apxc' and the session 'EmailMissionSpecialist:17@07/22 18:11:54'. The 'Heap' tab is active, showing three main sections: 'Types', 'Instances', and 'State'.

Types Table:

Type	Count	Total Size
EmailMissionSpecialist	1	4
List<Messaging.SendEmailR...	1	8
List<String>	1	8
Messaging.SendEmailResult	1	4
Messaging.SingleEmailMess...	1	16
String	3	166

Instances Table:

Address	Size
0x71cfdb96	16

State Table:

Field	Value
entityattachments	
ccaddresses	
plaintextbody	Mission Control 123, Thi...
emailpriority	Normal
usesignature	true
senderdisplayname	
subject	Flight Path Change
bccsender	false

References Table:

Inbound References	Referencing Instances		
Field	Type	Address	Size

Checkpoints Table:

Namespace	Class	Line	Date
none	EmailMissionSpecialist	17	07/22 18:11:54

Checkpoint Locations Table:

File	Line	Iteration
EmailMissionSpecialist	17	1

Symbols

Quick way to review the states of various objects at any checkpoint. This tab displays all symbols in memory using a tree view.

Execute SOQL and SOSL Queries