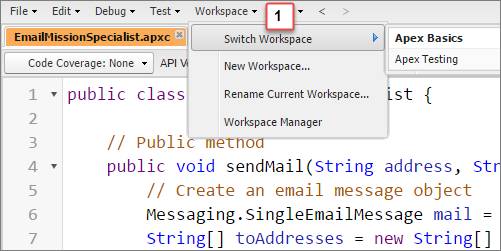
**Setup > Developer Console**

Workspace is just a way to organize resources

**Set up your own workspace**

**Select Workspace** | **New Workspace** and give the workspace a name. In the new workspace, you can create Apex classes, Aura components, Visualforce pages, and more.

If you want to switch between workspaces , choose Workspace | **Workspace Manager , Workspace** | **Switch Workspace, ( 1).**This way you can work with the code in each project or analyze the logs by simply opening another workspace.



To run an Apex class:

Debug > Open Execute Anonymous Window

**Lightning components:**

Create responsive mobile & desktop apps

Component bundles – act like folders that hold components and related resources such as stylesheets, controllers, designs, etc…

Aura components

File > New > Lightning Component

To access any resource:

File > Open Lightning Resources

Visualforce

Page-centric. Request to server, response to client

Lightning component framework is mostly client-side

Example: Visualforce Checkboxes

<apex:page sidebar="false">

<!--Flight Systems Checklist Visualforce Page-->

<h1>Checklist</h1>

<apex:form id="engineReadinessChecklist">

<apex:pageBlock title="Flight Systems Checklist">

<!--First Section-->

<apex:pageBlockSection title="Engines">

<!--Adding Checkboxes-->

<apex:inputCheckbox immediate="true"/>Engine 1

<apex:inputCheckbox immediate="true"/>Engine 2

<apex:inputCheckbox immediate="true"/>Engine 3

<apex:inputCheckbox immediate="true"/>Engine 4

<apex:inputCheckbox immediate="true"/>Engine 5

<apex:inputCheckbox immediate="true"/>Engine 6

</apex:pageBlockSection>

<!--Second Section-->

<apex:pageBlockSection title="Fuel Tanks">

<apex:inputCheckbox immediate="true"/>Tank 1

<apex:inputCheckbox immediate="true"/>Tank 2

<apex:inputCheckbox immediate="true"/>Tank 3

<apex:inputCheckbox immediate="true"/>Tank 4

<apex:inputCheckbox immediate="true"/>Tank 5

<apex:inputCheckbox immediate="true"/>Tank 6

</apex:pageBlockSection>

<apex:pageBlockButtons>

<!--Adding Save Button-->

<apex:commandButton value="Save" action="{!save}"/>

</apex:pageBlockButtons>

</apex:pageBlock>

</apex:form>

</apex:page>



Open a saved page:

File > Open > Pages

**LOGS**

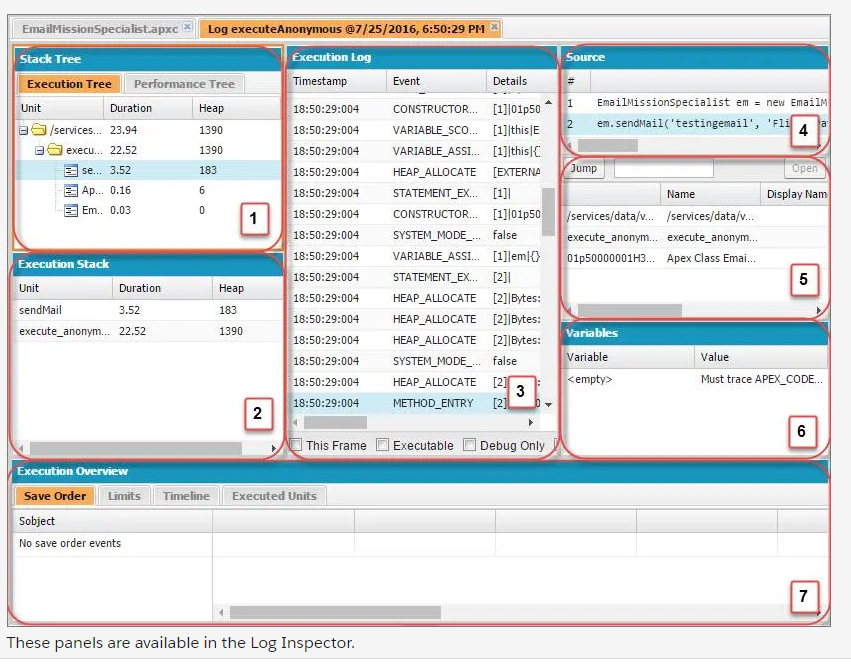
You can view your log in the following ways.

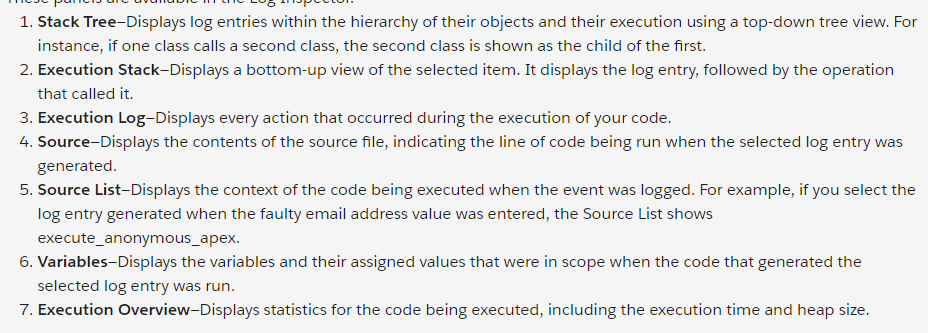
* **Before execution**, enable **Open Log** in the Enter Apex Code window. The log opens after your code has been executed.
* **After execution**, double-click the log that appears in the Logs tab.

**File** | **Open Raw Log**

**Debug** | **View Log Panels** – this is for large logs. Avail when you are viewing a debug log tab

System.debug('Your Label: ' + yourVariable);





Debug > Switch Perspectives or Perspective Manager: you can create your own perspective to view logs

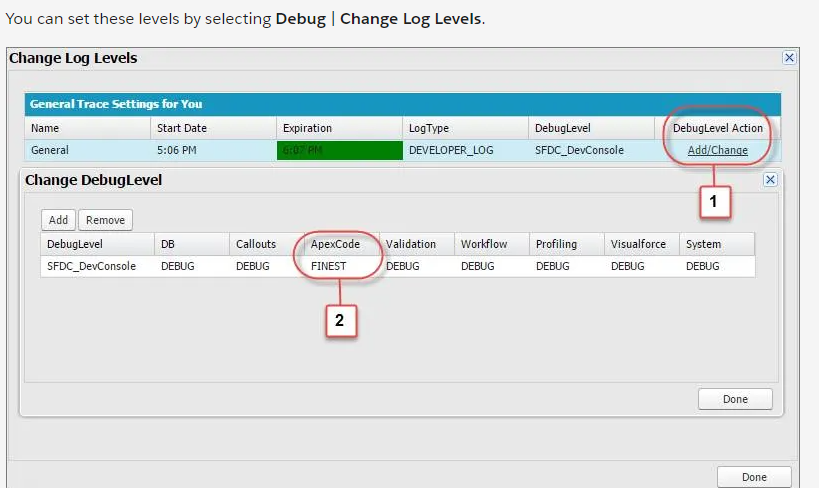
Categories

ApexCode

Database

**Log Levels**

* NONE
* ERROR
* WARN
* INFO
* DEBUG
* FINE
* FINER
* FINES



<https://help.salesforce.com/apex/HTViewHelpDoc?id=code_dev_console_tab_browser_logs.htm&language=en_US>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=code_dev_console_view_system_log.htm&language=en_US>

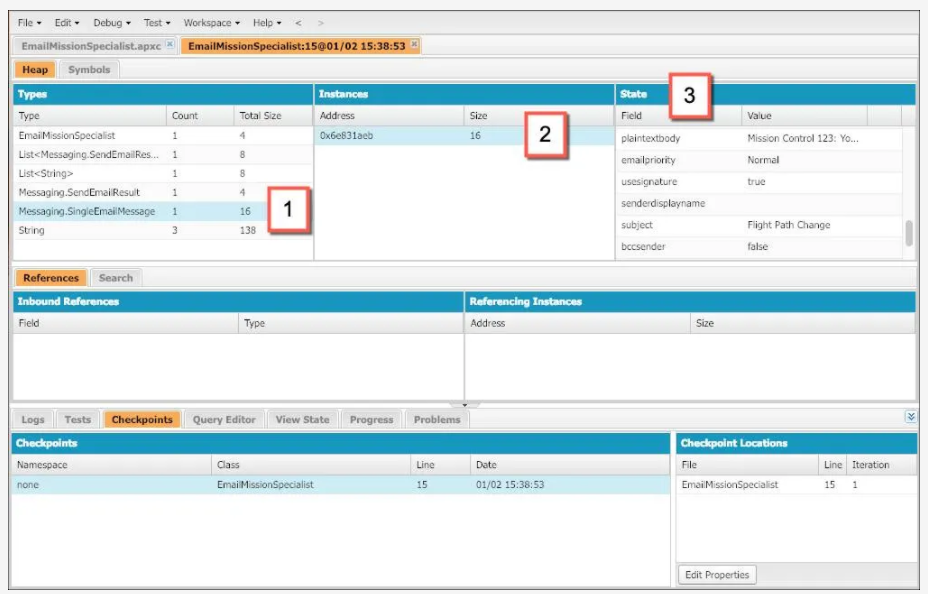
<https://help.salesforce.com/apex/HTViewHelpDoc?id=code_debug_log.htm&language=en_US>

**Checkpoints**

**Note:** To set checkpoints, you need the View All Data user permission. To generate results using checkpoints, run code using execute anonymous, or set a DEVELOPER\_LOG trace flag on yourself. The trace flag must have a log level for Apex of INFO or higher.

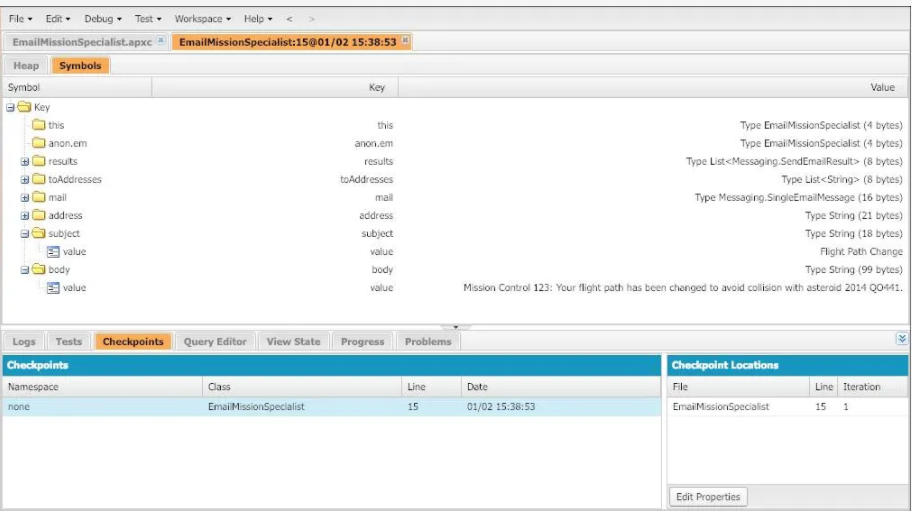
Breakpoints do not stop execution. After the run is complete, go to the Checkpoints tab to see the variable values when the breakpoint was hit

HEAP:



1. Under Types, click Messaging.SingleEmailMessage.
2. Under Instances, click any instance of this object type.
3. Under State, view the object’s fields and their values.

Symbols:



<https://help.salesforce.com/apex/HTViewHelpDoc?id=code_dev_console_tab_checkpoints.htm&language=en_US>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=code_dev_console_checkpoints_setting.htm&language=en_US>

<https://help.salesforce.com/apex/HTViewHelpDoc?id=code_dev_console_view_checkpoints.htm&language=en_US>

**SOSL – Salesforce Object Search Language**

FIND {Crisis} IN ALL FIELDS RETURNING Contact(FirstName, LastName, Phone, Email, Title)

Run in Query Editor

Or in the Debug > Open Execute Anonymous Window:

List<List<sObject>> searchList = [FIND 'Crisis' IN ALL FIELDS

RETURNING Contact(FirstName, LastName,

Phone, Email, Description)];

Contact[] searchContacts = (Contact[])searchList[0];

System.debug('Found the following contacts:');

for (Contact c : searchContacts) {

System.debug(c.LastName + ', ' + c.FirstName);

}