knowledge15
April 19–24, 2015 • Mandalay Bay • Las Vegas, NV

Lab Guide

Debugging 301: Problem Diagnosis and Prevention Techniques

Josh Addington & Andrew Venables

Login / Passwords:

admin / Knowledge15

itil / Knowledge15

employee / Knowledge15

This

Page

Intentionally

Left

Blank

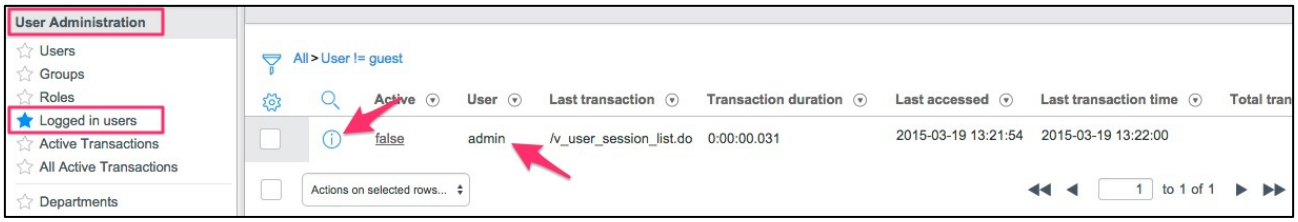
Lab Goal

This lab explains how to use the Log File Download and Log File Browser to detect slow and recursive business rules effecting performance issues.

Lab 1
Use App Server Logs to Find Issues

Use the Log File Utilities and App Log

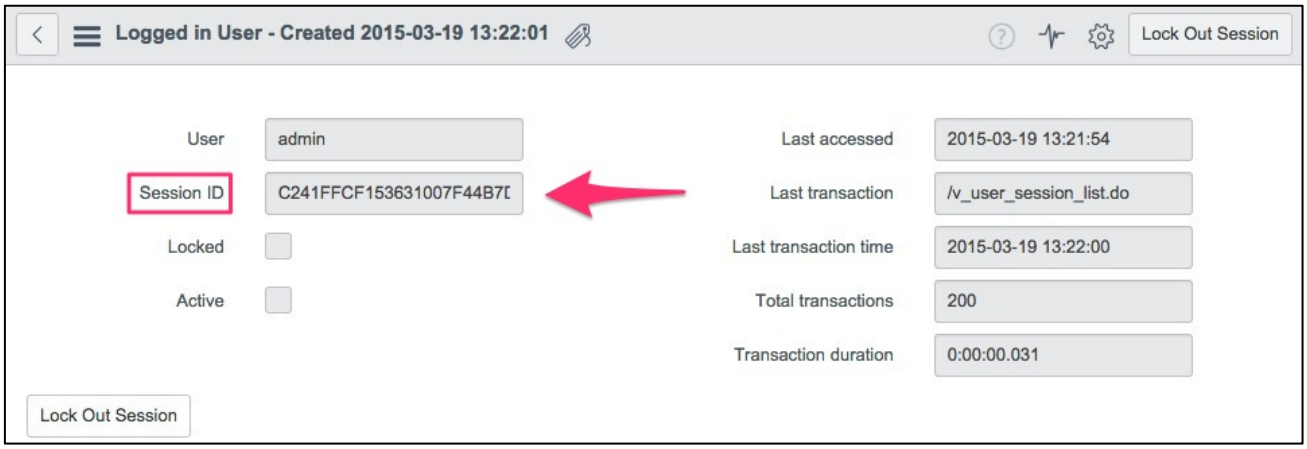
1. Login to the instance URL on the cover page using the provided **admin** credentials.
2. Navigate to **User Administration > Logged** in users.



3. Click the user record for the user for whom you are logged in.

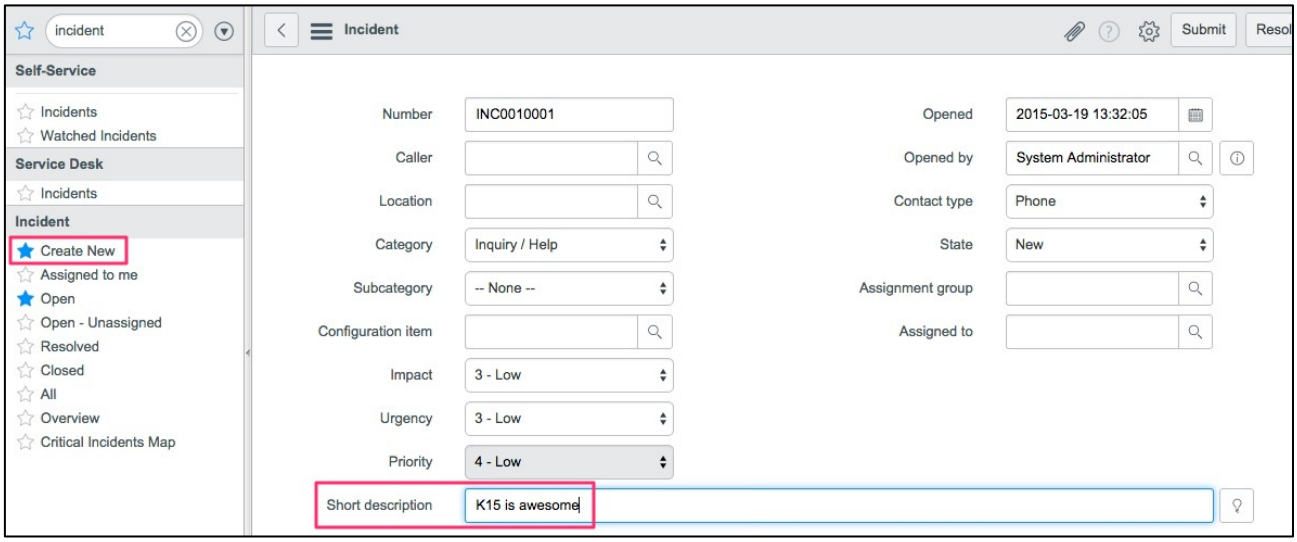
®2015 ServiceNow, Inc. All rights reserved.

4. Once you are in the logged in user record copy the Session ID. Keep it in your copy/paste buffer until needed later in this lab.



5. Navigate to **Incident > Create New**.

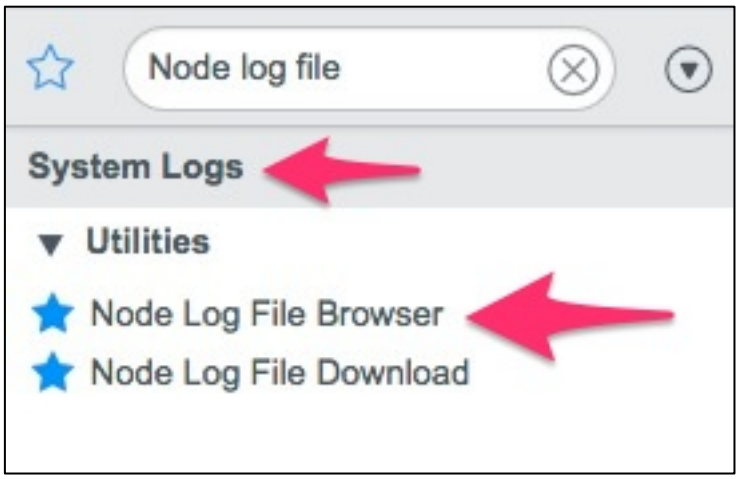
6. Fill out the **Short description** field on the form with something simple and then click **Submit**.



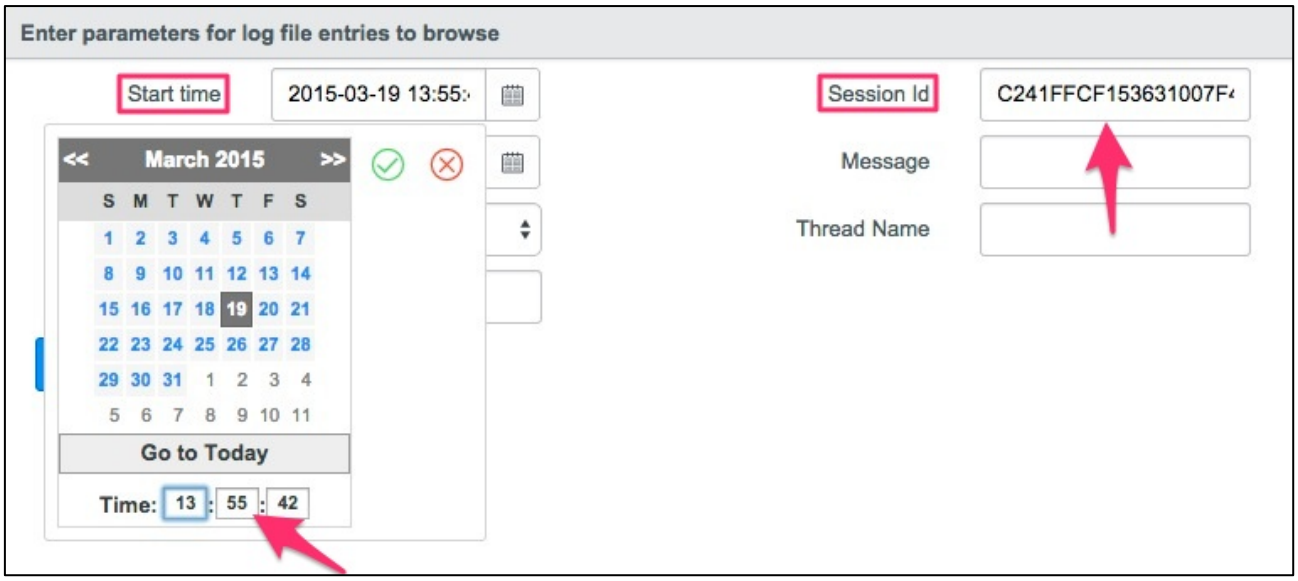
7. Allow for some time to pass for the incident to create (5-10 seconds). Intentionally, there are some business rules to slow down the process.

®2015 ServiceNow, Inc. All rights reserved.

8. Navigate to **System Logs > Log File Browser**.



9. Copy the Session ID in your buffer from step 4 above and paste it into **the Session Id** field. Also change the Start time field. Change the minutes to reflect a time before creating the incident in step 6. Leave the rest default.



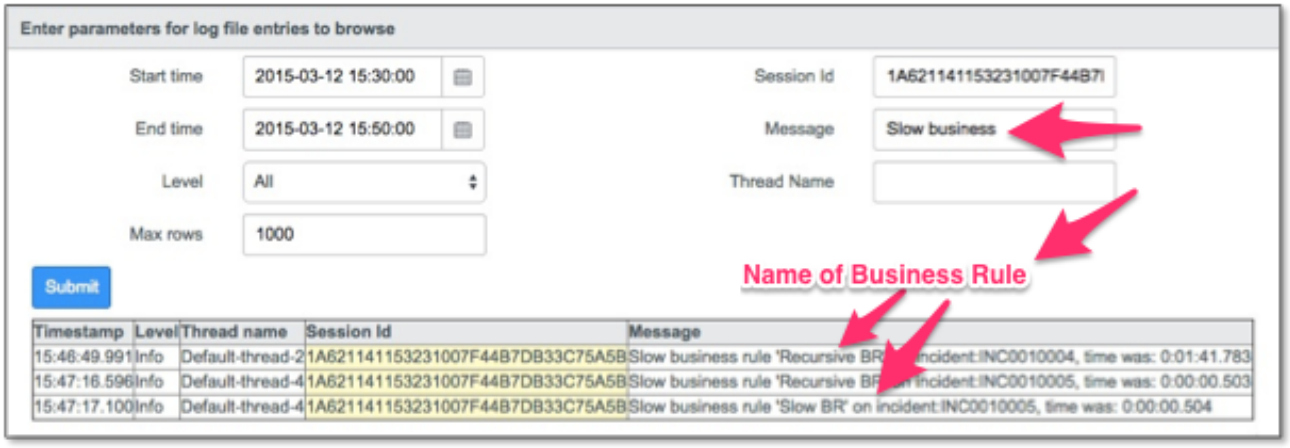
10. There is a log file dump for any given user session. Either search with the Browser search feature or filter further, using the **Message** field for the key words discussed previously in the presentation, **Slow business rule** or **Recursive business rule**.

11. Find those business rules from your incident form and disable them.

©2015 ServiceNow, Inc. All rights reserved.

Lab Success Verification

1. The screen should look like the example.



Challenge

1. Attempt to download the log file locally and either write a script or **grep** for the key words mentioned during the presentation. This can be something you do as a preproduction check on new code you introduce or if you see performance issues.

®2015 ServiceNow, Inc. All rights reserved.

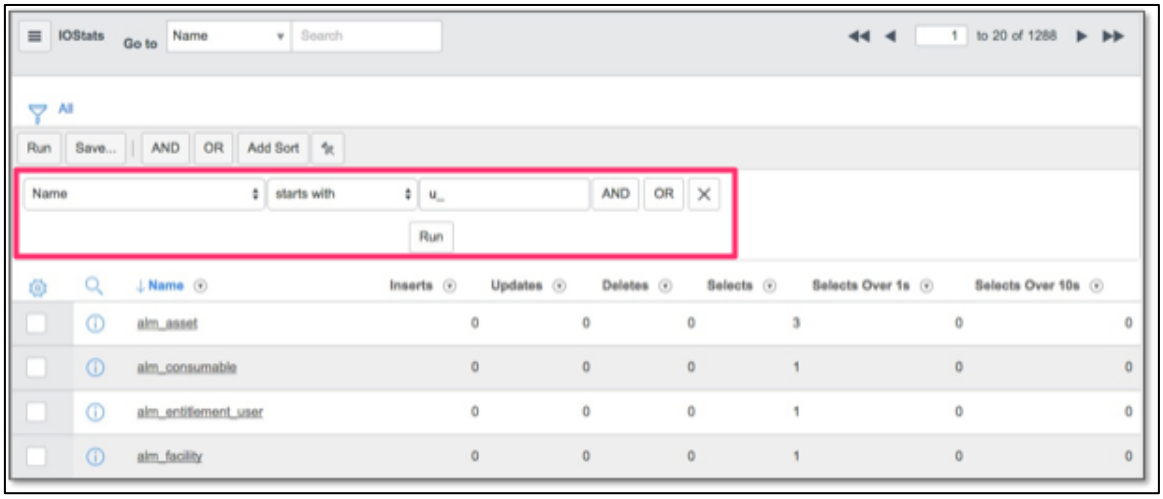
Lab Goal

This lab explains how to use the **IOStats** module to identify a script that is making too many SQL inserts.

Lab 2 Use IOStats

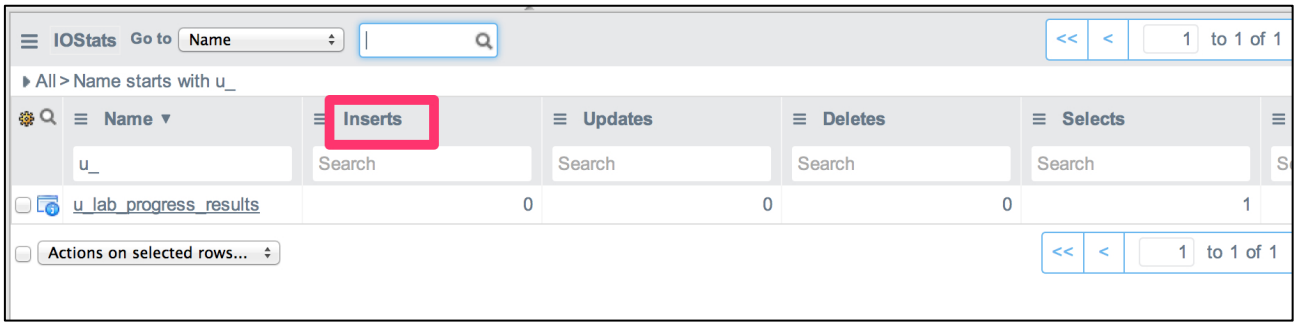
Use the IOStats Module

1. Log in to the instance URL on the cover page using the provided **admin** credentials.
2. Navigate to **System Diagnostics > Table IOStats**.
3. Change the filter to look for custom tables by searching for table names that start with **u_** (user defined table).



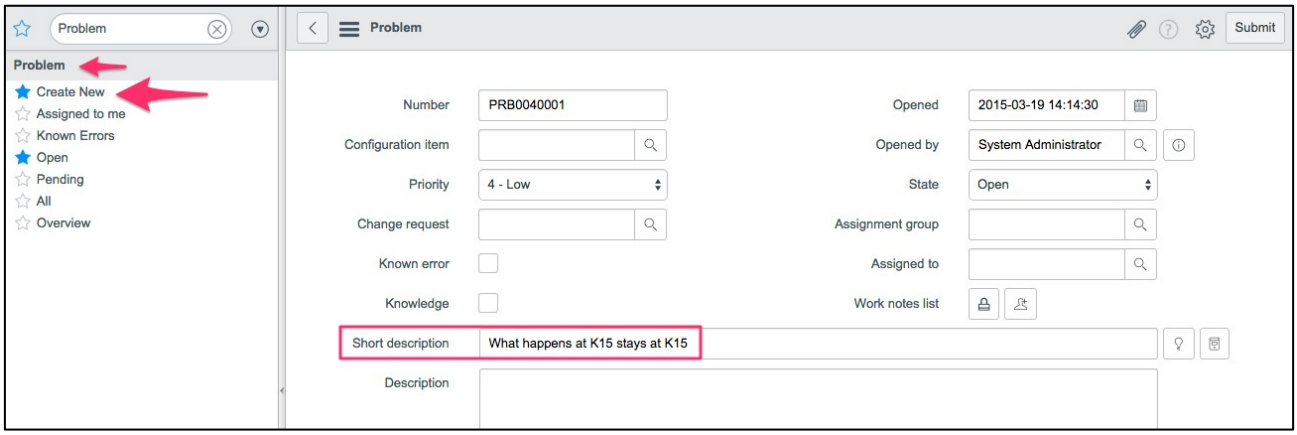
©2015 ServiceNow, Inc. All rights reserved.

4. Take note of the number of inserts that have occurred on these custom tables. Your values may differ than the example below, but you are interested in the change for this exercise.



5. Navigate to **Problem > Create New**.

6. Fill out the **Short Description** field on the form. Keep it simple. Click **Submit**.



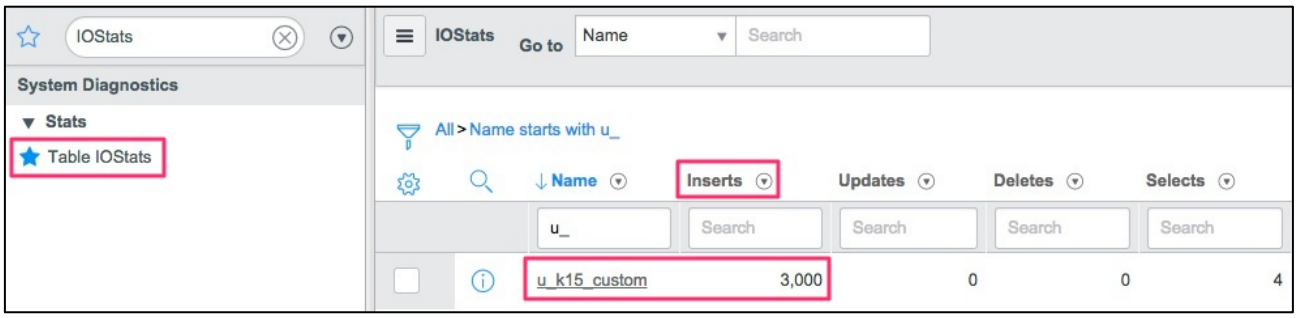
7. Repeat **Steps 5 – 6** two times, changing the **Short Description** each time.

8. Navigate to **System Diagnostics > Table IOStats**.

9. Change the filter to look for custom tables by searching for table names that start with **u_** (user defined table).

©2015 ServiceNow, Inc. All rights reserved.

10. Note that the number of inserts increased greatly when a problem was created.



11. This is a simple lab, but keep an eye on the count of DB transactions made (either over time or after an action). It is also helpful to monitor the selects over **1** second and **10** seconds. You can identify slow-running DB queries that way and use tricks like **setLimit()** when getting data.

12. Find the corresponding business rules from the problem table and disable them.

Challenge

1. Navigate to the **Slow Queries** stats in the **System Diagnostics Application**. This is a clean instance, but familiarize yourself with the columns you can sort. If you have a development or test instance, you can use instead try using this tool to identify slow queries.

©2015 ServiceNow, Inc. All rights reserved.

Lab Goal

This lab explains how to use the Event Log to find performance issues. Events are powerful, but they can impact system performance.

Lab 3
Event Log

Use the Event Log Module

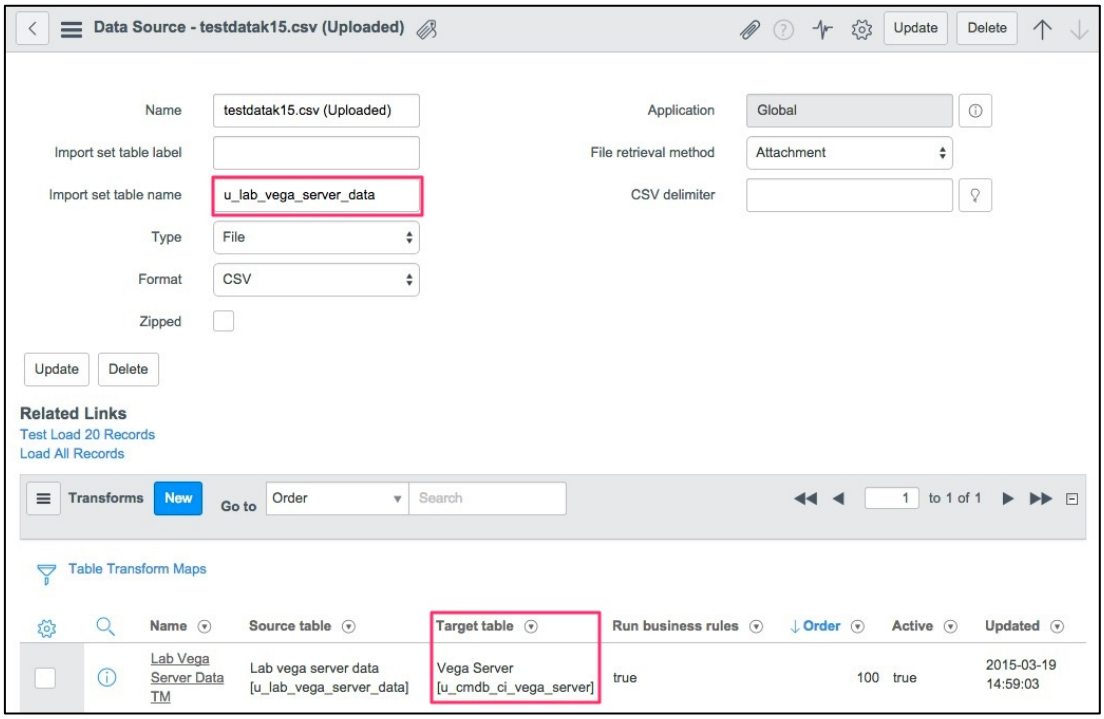
1. Log in to the instance URL on the cover page using the provided **admin** credentials.
2. Navigate to **Self-Service > Knowledge**.
3. A grid of each of the knowledge bases appears in the instance. Click the **Knowledge** knowledge base.
4. That's right. Fuji has multiple knowledge bases!



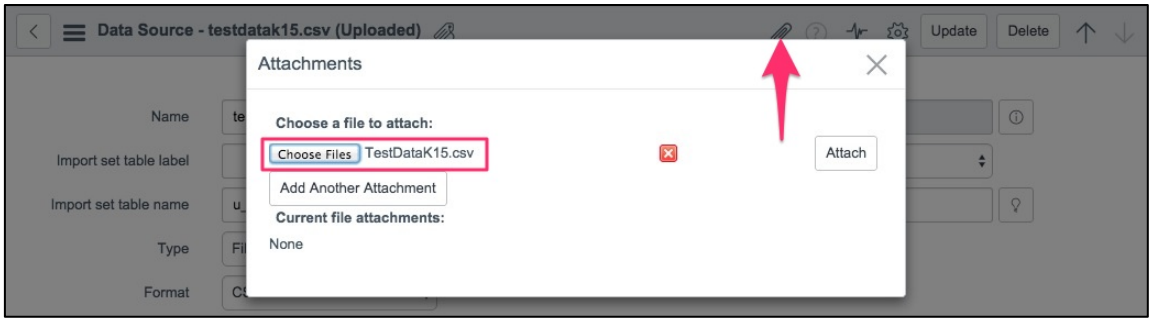
5. Select the **Lab3 Test Data** article and download the **TestDataK15.csv** file to your local machine. You attach this to the data source in the next steps.
6. Navigate to **System Import Sets > Data Sources**.

®2015 ServiceNow, Inc. All rights reserved.

7. Click on **the testdatak15.csv** data source. The import set table name should be **u_lab_vega_server_data**, and the target table should be **u_cmdb_ci_vega_server**.

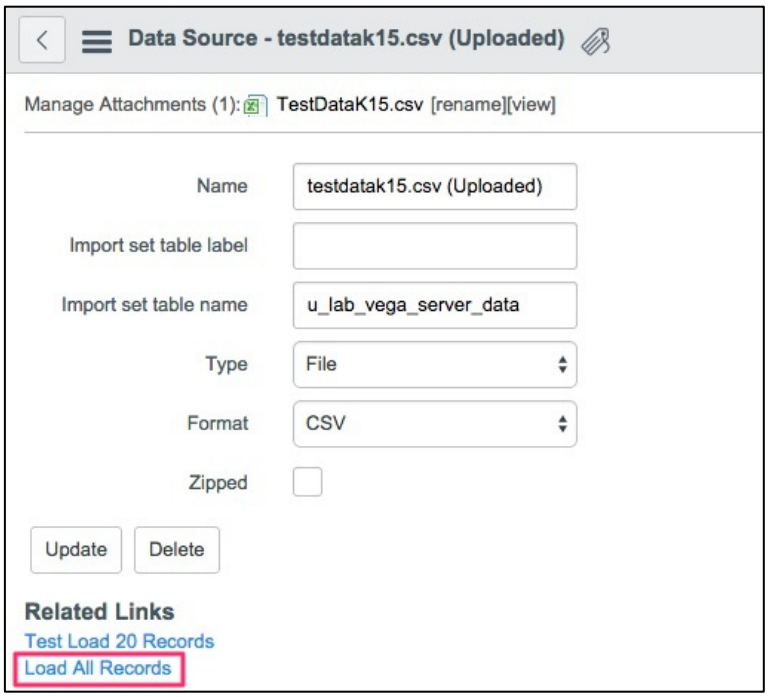


8. Click on the attachment icon and attach the file just downloaded from the Knowledge page.

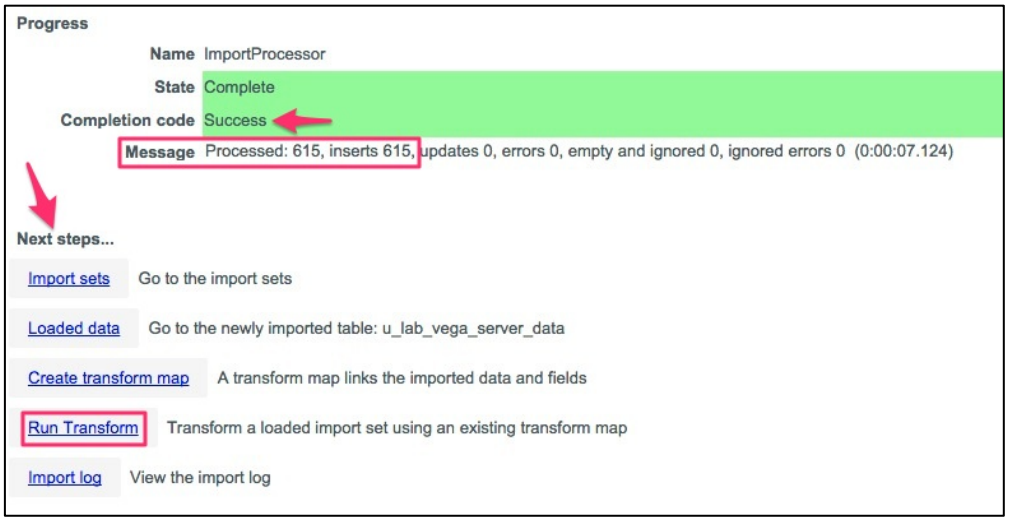


®2015 ServiceNow, Inc. All rights reserved.

9. Click the **Load All Records** related link.

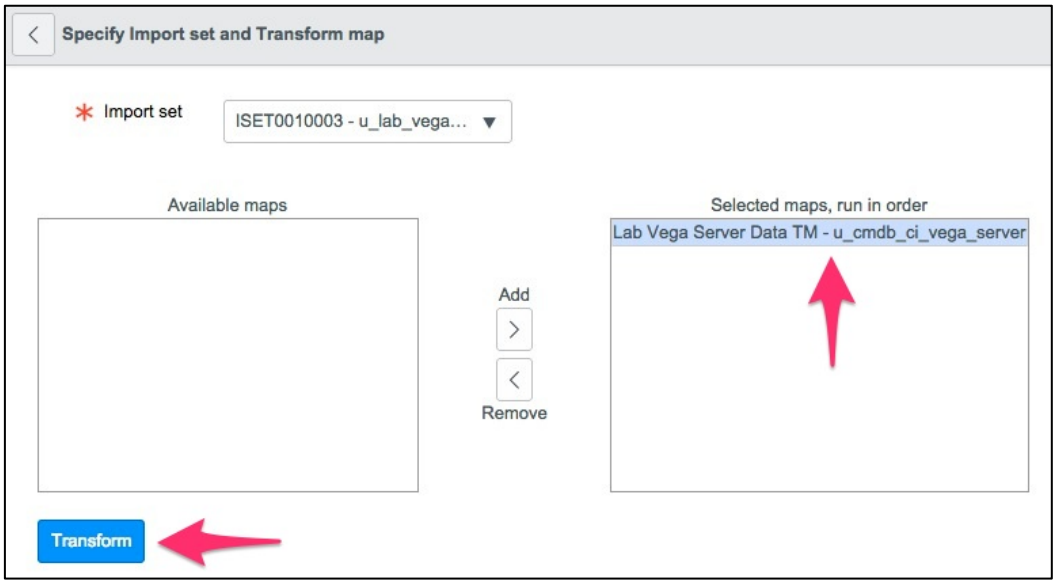


10. Make sure that **615** record inserts were made. Click the **Run Transform** link in the **Next steps…** section.

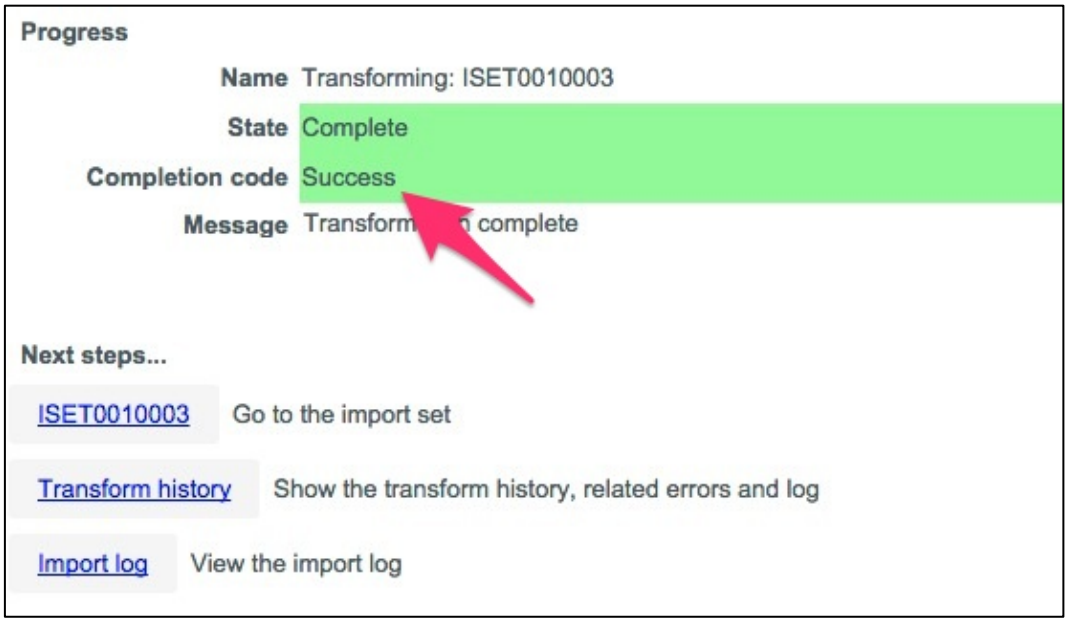


©2015 ServiceNow, Inc. All rights reserved.

11. The next screen asks you to select a transform map to use. There should only be one. Make sure it is on the **Selected** maps side of the slush bucket. Click **Transform**.



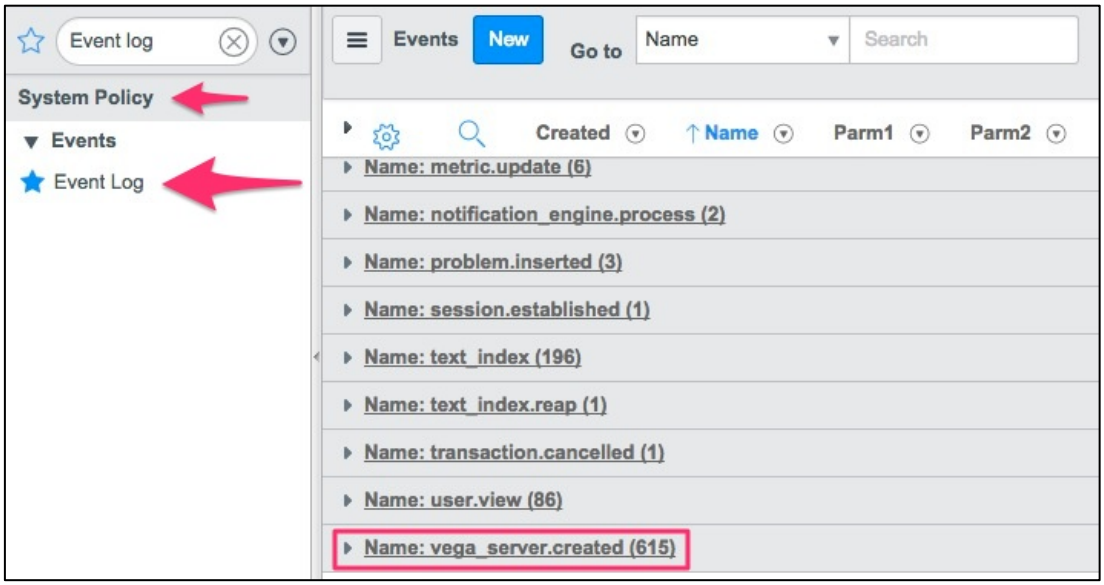
12. When the transform completes you should see a **Success** indication.



13. Navigate **to System Policy > Event Log**.

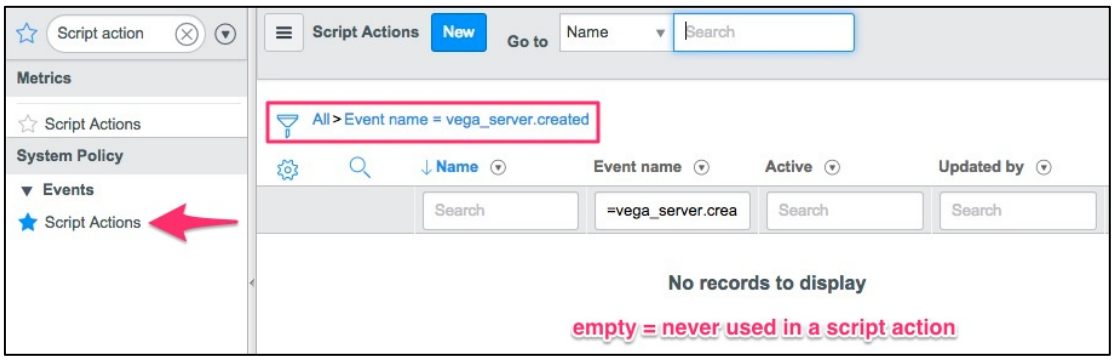
®2015 ServiceNow, Inc. All rights reserved.

14. Right-click the **Name** column of the Event Log and select **Group-By Name**.



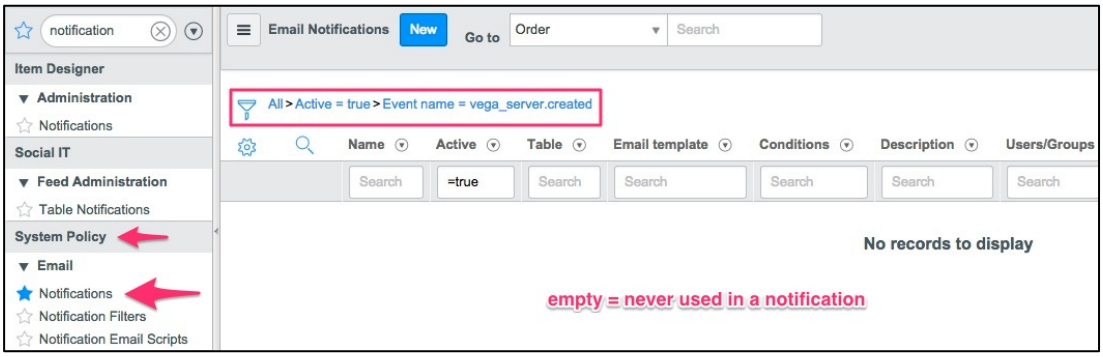
15. You can identify clearly that there is an event firing quite a lot. You need to figure out what it does. Take note of the name **vega_server.created**.

16. There are two main places that events are handled. First the Script Action module under **System Policy > Events**. Filter search where event name is **vega_server.created**.



©2015 ServiceNow, Inc. All rights reserved.

17. The second place an event can be handled is the email notification. Navigate to **System Policy > Notifications** and filter search for active notifications and Event name is **vega_server.created**.



18. You identified an event that is firing when you upload data. It is filling up the event backlog, but not doing anything. Deactivate the business rule firing this event.

Lab Success Verification

1. You have successfully completed this lab if you can identify the business rule that is firing the event and you deactivate it.

®2015 ServiceNow, Inc. All rights reserved.

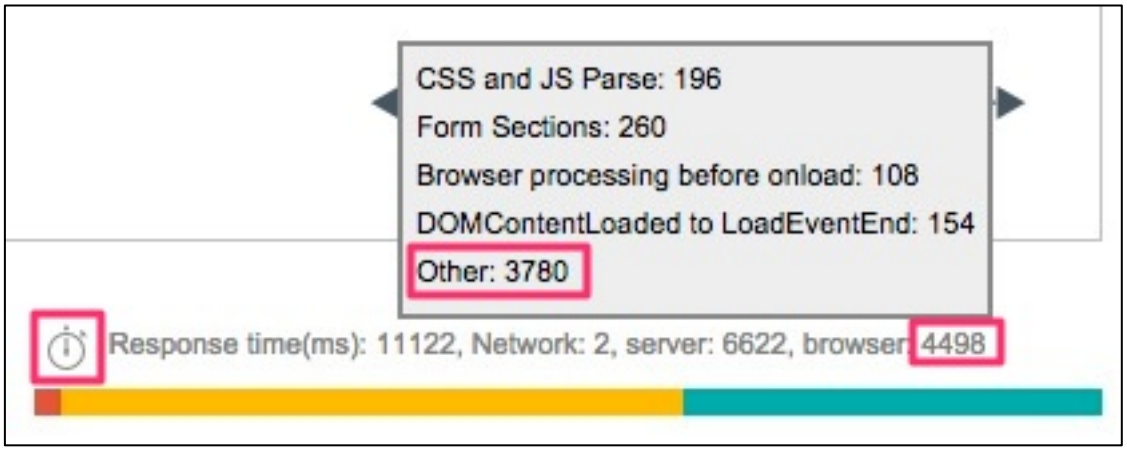
Lab Goal

This lab explains how to refactor a client script to make it maintainable, more readable, and better performing.

Lab 4 Optimize the Client Side

Use GlideAjax, g_scratchpad, and GSLog

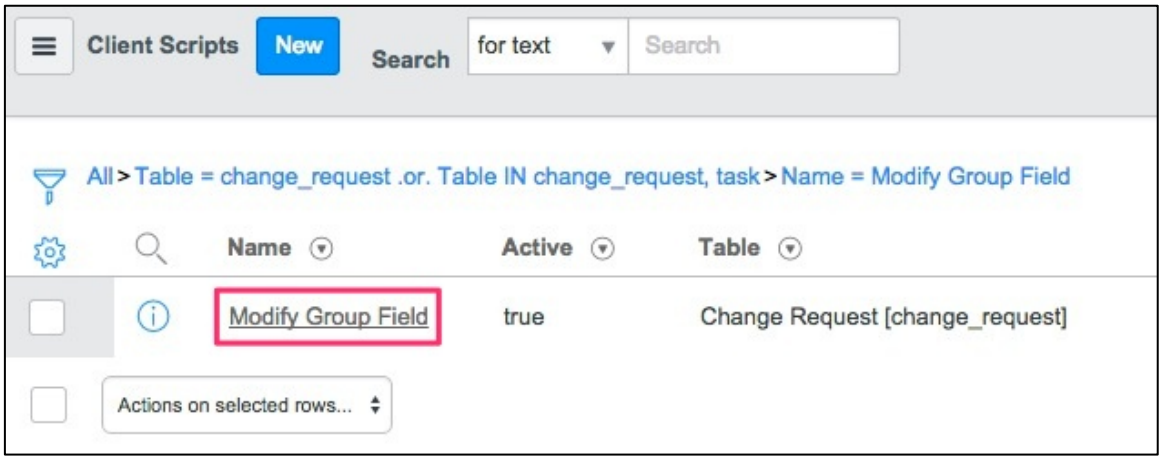
1. Log in to the instance URL on the cover page using the provided **admin** credentials.
2. Navigate to **Change > Open > CHG0000012**.
3. Scroll to the bottom of the page and investigate the browser transaction timings by clicking on the response time icon ( ).



4. Find the client-side script that is taking a long time to load.

©2015 ServiceNow, Inc. All rights reserved.

5. Navigate to **System Definition > Client Scripts** or right-click the change header and **Configure > Client Scripts**.



®2015 ServiceNow, Inc. All rights reserved.

6. Open the script **Modify Group Field** on the table **change_request** and replace it with the script below.

```
function onChange(control, oldValue, newValue, isLoading, isTemplate)
{
   //do nothing if the form is loading and the group is empty
   if ((isLoading && oldValue=="") || newValue == '') {
      return;
   }
   //if the group is filled onLoad, fill the member field
   else if(isLoading && oldValue!="")
   {
         g_form.clearOptions("u_group_member");
         fillGroupMembers(g_scratchpad.chg_groupmembers);
   }
   else //retrieve the value from
   {
      var ga = new GlideAjax('ChangeManagementUtils');
      ga.addParam('sysparm_name','getGroupMembersOnDemand');
      ga.addParam('sysparm_assignement_group', newValue);
      ga.getXML(processResponse);
      g_form.clearOptions("u_group_member");
   }
}
//process the ajax call
function processResponse(response)
{
   var answer =
response.responseXML.documentElement.getAttribute('answer');
   if (answer)
   {
            fillGroupMembers(answer);
   }
}
```

```
//function fill the group members field with the options
function fillGroupMembers(members)
{
   members = members.evalJSON(true);
   for(member in members)
   {
         g_form.addOption("u_group_member", member, members[member]);
   }
}
```

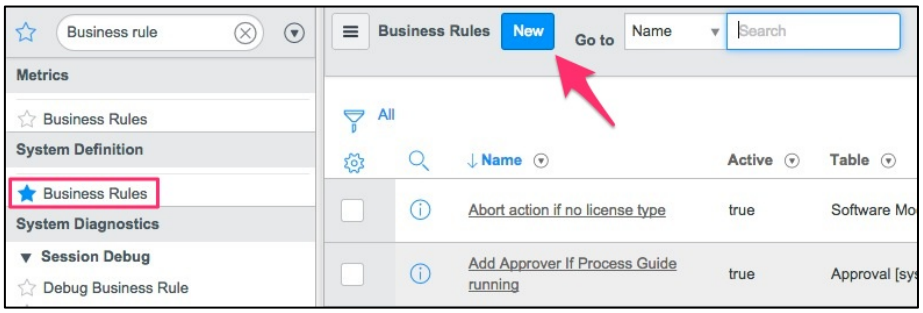
The script is in your lab instance in the Knowledge Base Article **KB0010002**.

©2015 ServiceNow, Inc. All rights reserved.

Results:

OnLoad: Only if the assignment group field is filled will the script be triggered and the values from the **onDisplay** business rule used.
onChange: A single round trip to the server has been realized by using **GlideAjax**.
The script is optimized to its best and well documented for further enhancements.

7. Navigate to **System Definition > Business Rules** and create a new one.



8. Set up the following:

Name: **Check for Assignment Group**
Table: **Change Request [change_request]**
When: **display**
Condition: **!current.assignment_group.nil()**
Script:

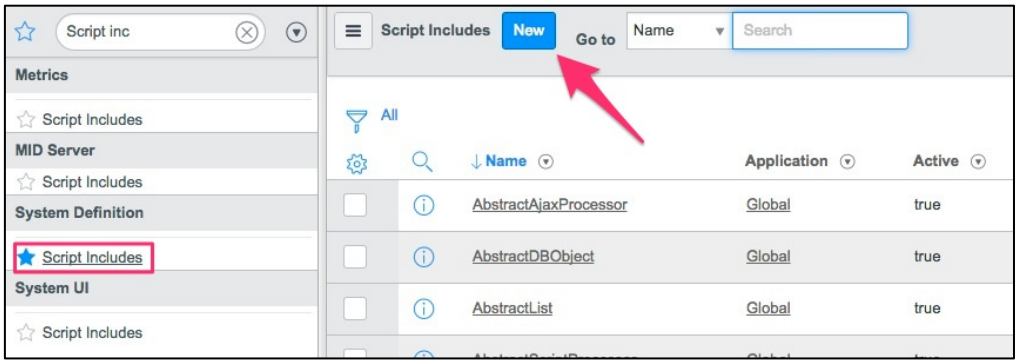
```
function onDisplay(current, g_scratchpad) {
    //This function will be automatically called when this rule is processed.
    var utils = new ChangeManagementUtils();
    var json = utils.getGroupMembers(current.assignment_group);
    g_scratchpad.chg_groupmembers = json; //return array
}
```

The script is in your instances in the Knowledge Base Article **KB0010003**.

Result: The values for the members will be prepopulated in the scratchpad and sent to the client but only in the event that an assignment group exists on the ticket.

©2015 ServiceNow, Inc. All rights reserved.

9. Navigate to **System Definition > Script Includes** and create a new one.



10. Set up the following:

Name: **ChangeManagementUtils**
Description: **Client & Server Side functions for CHG**
Client callable: **true**
Script:

```
var ChangeManagementUtils = Class.create();

ChangeManagementUtils.prototype =
Object.extendsObject(AbstractAjaxProcessor, {

//wrapper for the AjaxCall
  getGroupMembersOnDemand: function ()
  {
     var group = this.getParameter("sysparm_assignement_group");
     return this.getGroupMembers(group);
  },
```

©2015 ServiceNow, Inc. All rights reserved.

```
//retrieve all member by name and sysid for a given group
  getGroupMembers: function (group_sysid)
  {
       var json = new JSON();
       var result = {};

      //get the relevant group member
      var grMember = new GlideRecord("sys_user_grmember");
      grMember.addQuery("group", group_sysid);
      grMember.query();

      //build or condition to get the names and ids of the others
      if (grMember.next())
      {
            var grUsers = new GlideRecord('sys_user');
            var qc = grUsers.addQuery('sys_id',
grMember.getValue("user"));
          //add the other users to the query
            while (grMember.next())
            {
                  qc.addOrCondition('sys_id',
grMember.getValue("user"));
            }
            grUsers.query();
            //build JSON and put it into the scratchpad
            while(grUsers.next())
            {
                  result[grUsers.getValue("sys_id")] =
grUsers.getValue("name");
            }
            return json.encode(result); //return array
      }
  },

//logging functionality for all your Ajax calls
  _log: function (msg)
  {
        if (typeof this.logger == "object") //if exists
        {
            this.logger.info(msg);
        }
        else //if not exists, init
        {
            this.logger = new GSLog("mycompany.process.ajax.logging",
"ChangeManagementUtils");
              this.logger.info(msg);
        }
  },

  type: "ChangeManagementUtils"
});
```

®2015 ServiceNow, Inc. All rights reserved.

The script is on your lab instance in the Knowledge Base Article **KB0010003**.

Results: The script include reuses code and offers in addition a function to for logging. In case of an error, the logging can be switched on and off easily without interfering with other parts of the system.

Lab Success Verification

1. You have successfully completed this lab if you can identify the performance increase when loading the Change Management form.

©2015 ServiceNow, Inc. All rights reserved.

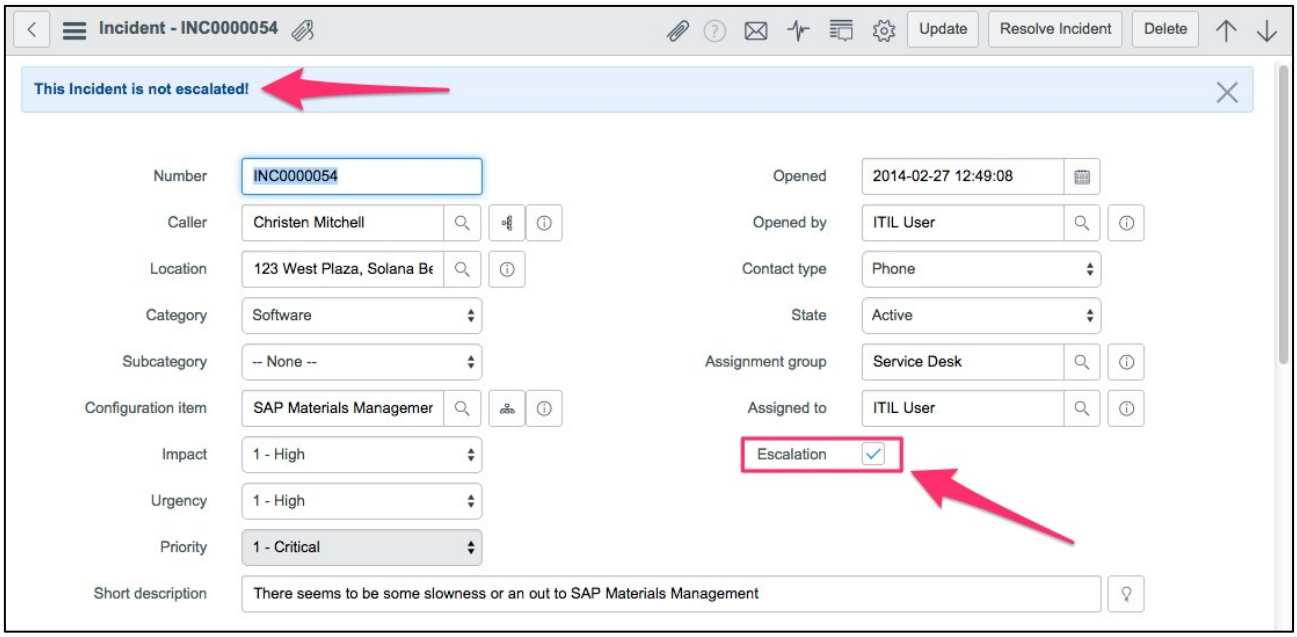
Lab Goal

This lab explains how to use the built-in browser debugger to fix a client script.

Lab 5
Debug the Client Side

Use a Debugger

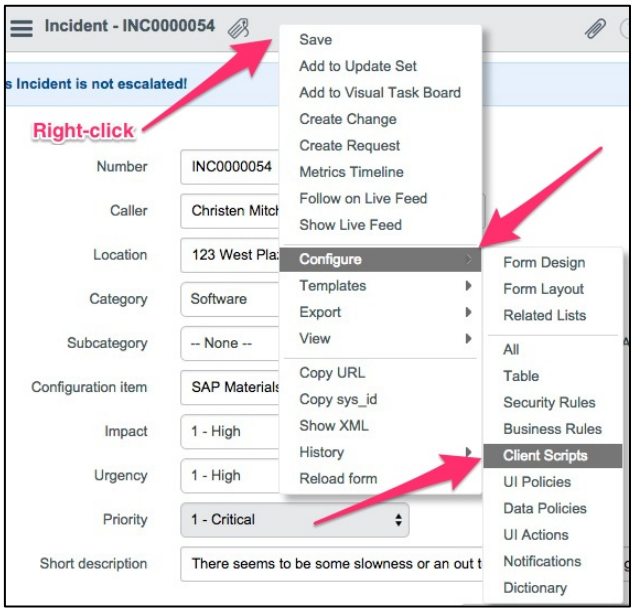
1. Use Chrome.
2. Login to the instance URL on the cover page using the provided **admin** credentials.
3. Navigate to **Incident > Open > INC0000054**.
4. Look at the top of the content frame. A message appears.



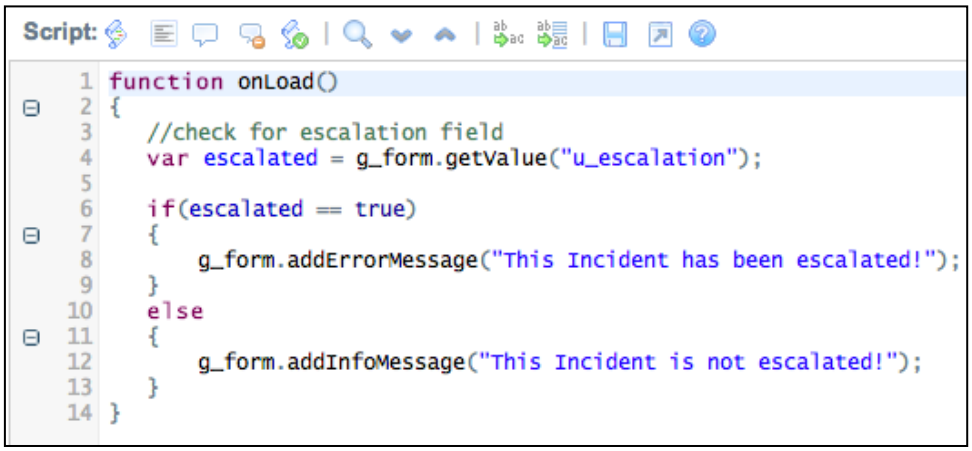
5. Someone escalated this Incident because the **Escalation** field is set to **true**. The message therefore is wrong and must be fixed.

©2015 ServiceNow, Inc. All rights reserved.

6. Navigate to **System Definition > Client Scripts** or right-click the incident header and select **Configure > Client Scripts** and open **Check Escalation** on the incident table

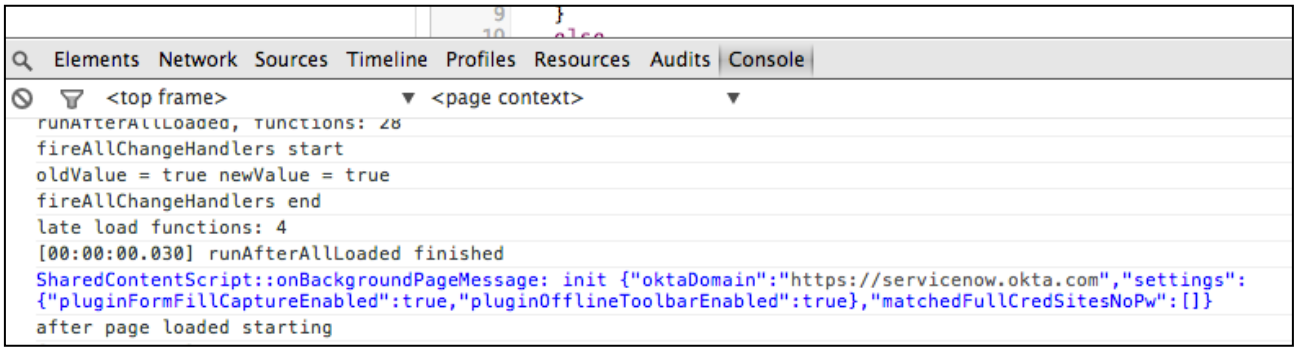


7. The script checks **onLoad** for the value of the **Escalation** field and sets a message accordingly. For **INC0000054**, the message should be **This incident has been escalated!**



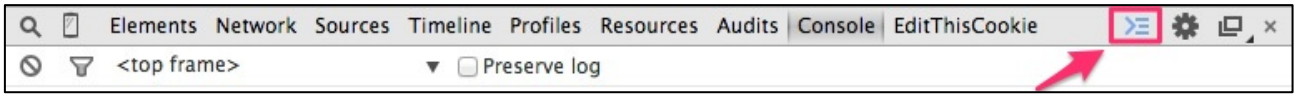
©2015 ServiceNow, Inc. All rights reserved.

8. Open **Chrome DevTools** by clicking **CMD + Alt + i**. The DevTools now appear on the bottom of the screen.



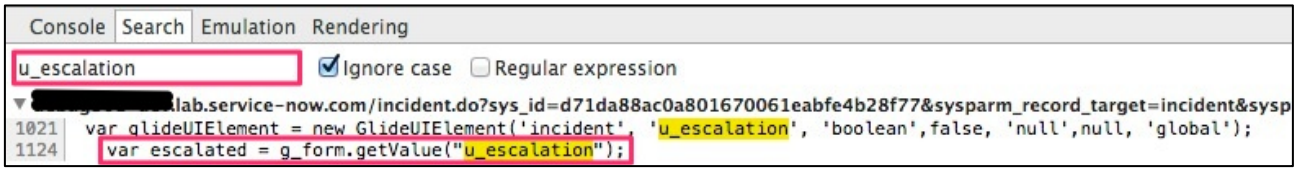
9. Navigate back to **Incident > Open > INC0000054**.

10. Press **ESC** or click on the icon ( ) on the upper right of the DevTools.



11. In the new bar that appears you can search the whole site for any content. In this case, you want to find the **onLoad** client script to set breakpoint and debug it.

    Click on **Search** and type in the field name **u_escalation**. Two results return. Take a closer look at the second one, which seems to be the line of the client script.



    Note: Your line numbers may be different.

©2015 ServiceNow, Inc. All rights reserved.

12. Click the second search result. Note the correct script popping up in the background. Press **ESC** again to close the search screen.



Note: Your line numbers may be different.

13. At this point you know that the comparison (**if-statement**) goes wrong. You assume that the variable **escalated** delivers the wrong values.

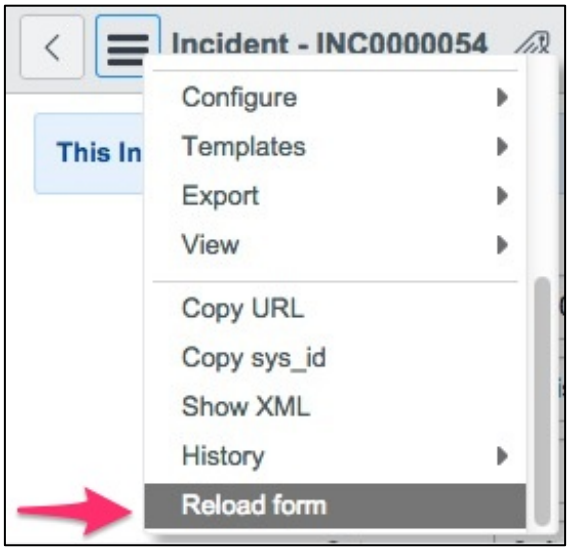
    To double check that you have to set a breakpoint at this line by clicking the line number in front of it. It is possible to set multiple breakpoints. The browser stops executing at this point and the debugger takes over.



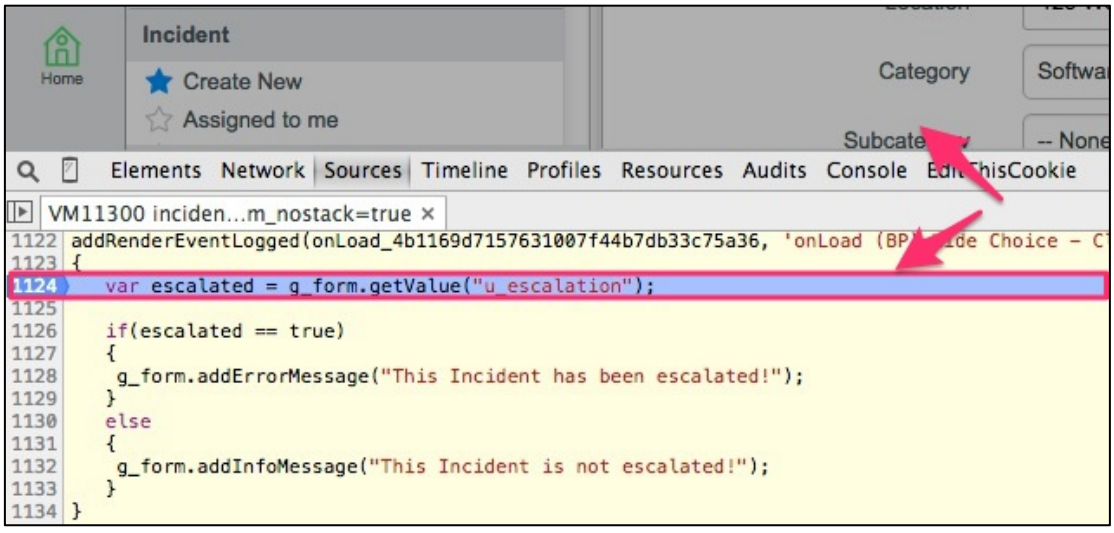
Note: Your line numbers may be different.

©2015 ServiceNow, Inc. All rights reserved.

14. Reload the **INC** by clicking the header bar and pressing **Reload Form**.



15. The browser now stops at the breakpoint. You can control the behavior. An indicator is that the line is colored in blue and the screen locked in gray.

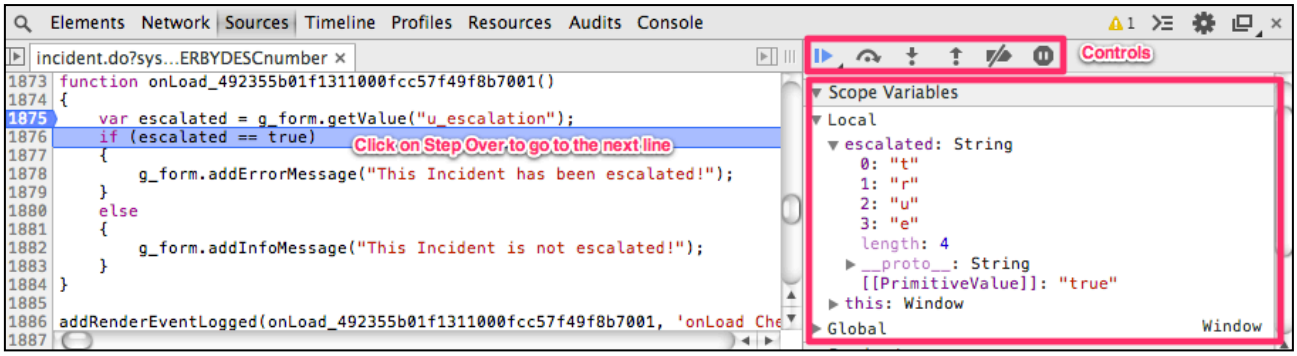


Note: Your line numbers may be different.

©2015 ServiceNow, Inc. All rights reserved.

16. Press **Step Over** ( ) to go to the next line in the code. On the top-right side find the controls, which allow you to step over (evaluate) to the next line in the code, step in to functions, step out of functions, or stop the debugging.

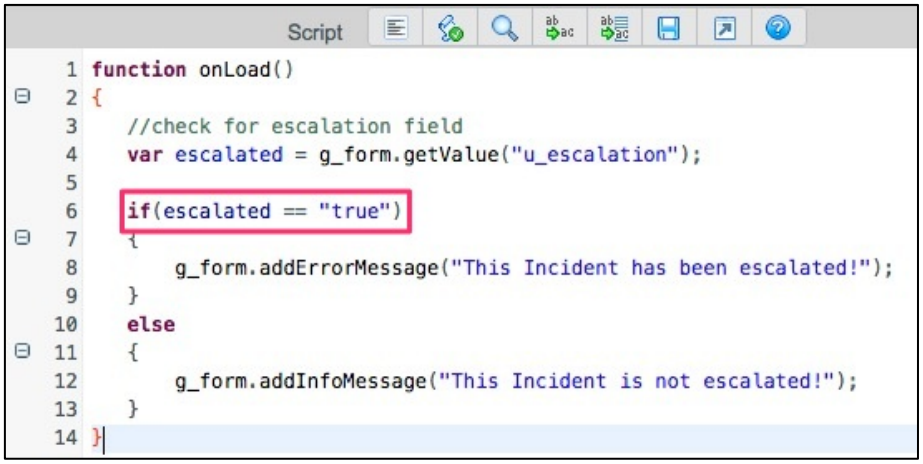
    In addition, you can investigate below the **Scope Variables**. This box shows all the initiated and available variables in the scope of the function.



17. In the **Scope Variables** box, note that the variable **escalated** is filled by the **getValue()** function with the value **true** for type String.

18. The **if** statement checks for the Boolean value **`true`**, which is not the same as the string value **true**. ***This is the reason that the script is giving the incorrect result*.**

19. Fix the script by modifying the script to match the example.



Results: The script now works properly because the **if**-statement can evaluate against the correct type of variable.

©2015 ServiceNow, Inc. All rights reserved.

Lab Success Verification

1. Open the Incident again and see the message properly changing according to the field **Escalation**.

®2015 ServiceNow, Inc. All rights reserved.