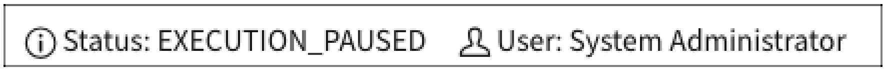
ServiceNow Application Developer

Server-side Scripting > Breakpoints

Breakpoints pause script execution to give developers the chance to examine variables and their values. The JavaScript Debugger status bar indicates whether scripts are paused at breakpoints or waiting for breakpoints to be reached.



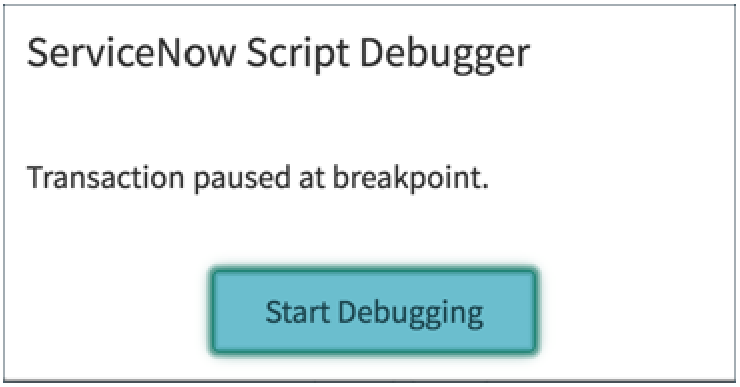
ⓘ Status: EXECUTION_PAUSED   User: System Administrator

Setting and Using Breakpoints

Set breakpoints in the JavaScript Debugger or in the Script Editor in a server-side script record. To set breakpoints, click in the gutter for the line of interest.

Breakpoints are indicated by blue arrows (5). Click an existing breakpoint to remove it.

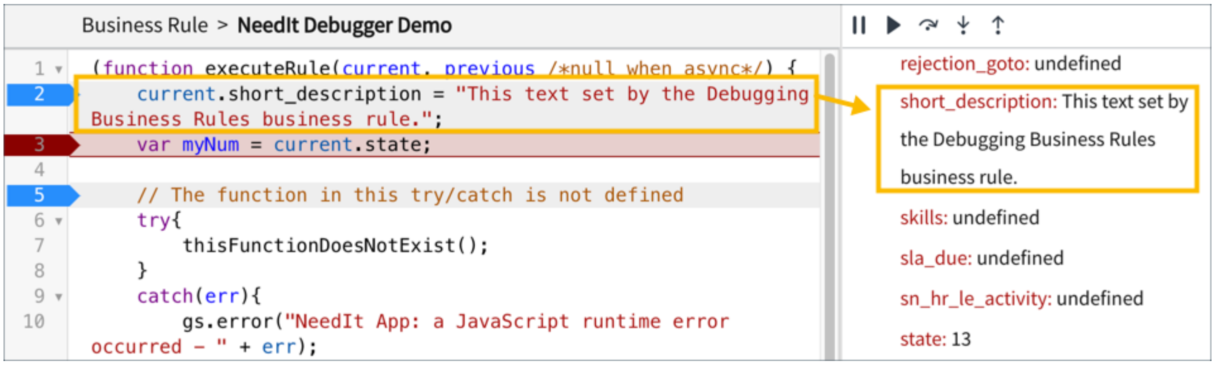
After setting breakpoints, force the script to execute by creating or modifying a record that meets the script trigger criteria. In the ServiceNow browser window, a dialog indicates that script execution is paused at a breakpoint.



ServiceNow Script Debugger

Transaction paused at breakpoint.

In the JavaScript Debugger, a red arrow and red highlight indicate that a script is paused at a breakpoint. While script execution is paused, examine the variables and their values. In the screenshot, script execution is paused at line 6. The logic on line 6 is not yet executed. The *short_description* value shown is from the *current* object. The *previous* object (not shown) has no values because the script is executing for a new record.



```
1  (function executeRule(current, previous /*null when async*/) {
2      current.short_description = "This text set by the Debugging
   Business Rules business rule.";
3      var myNum = current.state;
4
5      // The function in this try/catch is not defined
6      try{
7          thisFunctionDoesNotExist();
8      }
9      catch(err){
10         gs.error("NeedIt App: a JavaScript runtime error
   occurred - " + err);
```

rejection_goto: undefined
short_description: This text set by the Debugging Business Rules business rule.
skills: undefined
sla_due: undefined
sn_hr_le_activity: undefined
state: 13

When script execution is paused, use the controls to move through scripts:

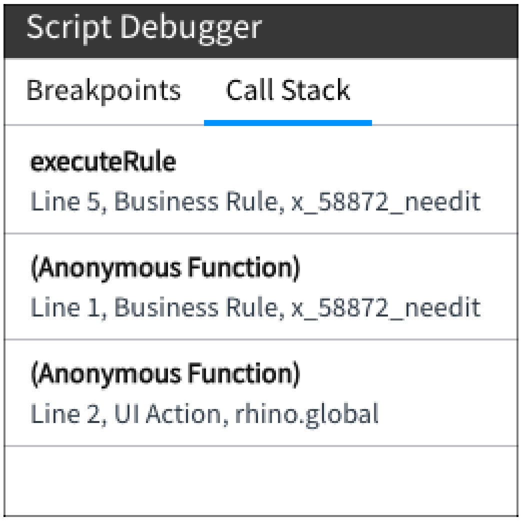
- ***Pause debugging*** (⏸): Stops any current debugging session, and disables the Script Debugger for the current user. The Script Debugger does not pause on breakpoints for the current user until it is restarted.
- ***Start debugging*** (⏻): Enables the Script Debugger for the current user. The Script Debugger pauses on breakpoints.
- ***Resume script execution*** (▶): Advances from the current breakpoint to the next breakpoint. If there are no other breakpoints, the script runs to completion.
- ***Step over next function call*** (↷): When execution is paused on a line that is about to invoke a function, step over causes the code in the function to execute, but does not pause on each individual line of code inside the function.

- ***Step into next function call*** (↓): When the Script Debugger pauses on a method call, this control allows the user to advance to the first line of executed code within the method call. Stepping into a method updates the current position within the call stack. If the user does not have read access to the method call, then this control acts like step over instead.

- ***Step out of current function*** (↑): When the Script Debugger pauses within a method call, this control allows the user to exit the current method call and return to the calling script from the call stack. If the user is not within a method call, then this control acts like step over instead.

Call Stack

The call stack shows list of methods and functions called to run as part of the script execution. Click an item in the call stack to see the definition. Execution must be paused at a breakpoint to examine the call stack.



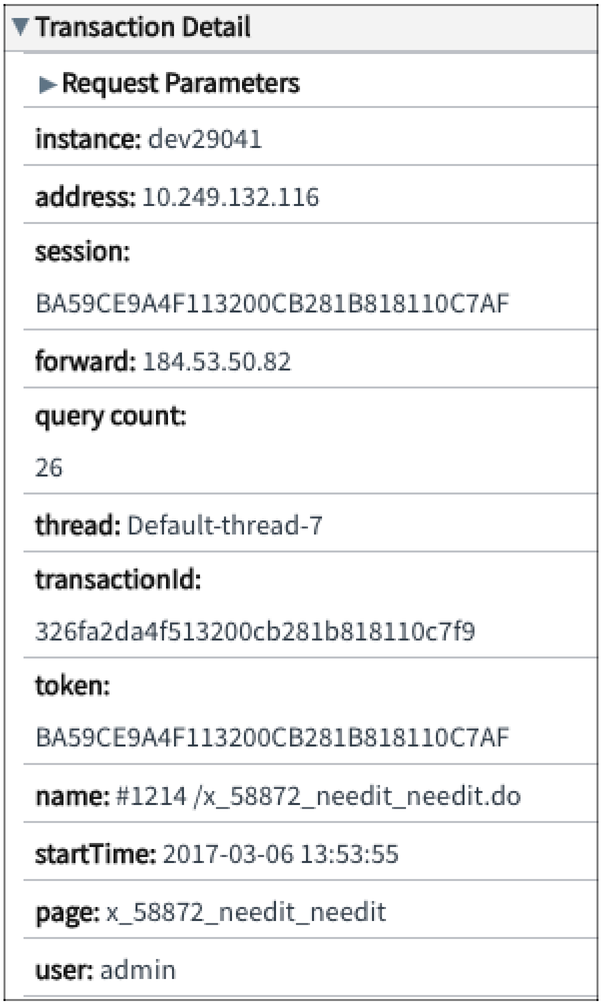
executeRule
Line 5, Business Rule, x_58872_needit

(Anonymous Function)
Line 1, Business Rule, x_58872_needit

(Anonymous Function)
Line 2, UI Action, rhino.global

Transaction Details

When paused at a breakpoint, the Transaction Details show information about the current transaction including the session ID, user, request parameters, instance, start time, and more.



▼ **Transaction Detail**

▶ **Request Parameters**

instance: dev29041

address: 10.249.132.116

session: BA59CE9A4F113200CB281B818110C7AF

forward: 184.53.50.82

query count: 26

thread: Default-thread-7

transactionId: 326fa2da4f513200cb281b818110c7f9

token: BA59CE9A4F113200CB281B818110C7AF

name: #1214 /x_58872_needit_needit.do

startTime: 2017-03-06 13:53:55

page: x_58872_needit_needit

user: admin