

ServiceNow Application Developer

Server-side Scripting > Debugging Business Rules

So far in this module, debugging has primarily been:

- Use the Script Editor to find JavaScript syntax errors
- Examine scripts to look for errors
- Verify the Business Rule is configured correctly

The strategies used so far are useful but are inadequate for fully debugging. Additional debugging strategies are:

- System Logs
- Debug Business Rule (Details)
- JavaScript Debugger

System Logs

The scoped *GlideSystem* API has logging methods:

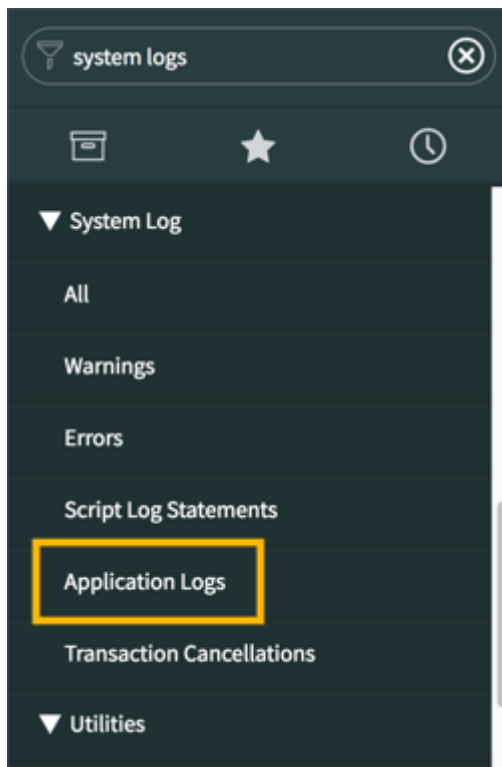
- ***gs.info()***(https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=r_ScopedGlideSystemInf_String_Object_Object_Object_Object_Object)
- ***gs.warn()***(https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=r_ScopedGlideSystemWarn_String_Object_Object_Object_Object_Object)
- ***gs.error()***(https://developer.servicenow.com/app.do#!/api_doc?v=madridn&id=r_ScopedGlideSystemError_String_Object_Object_Object_Object_Object)

- [gs.debug\(\)](https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=r_ScopedGlideSystemDebug_String_Object_Object_Object) (https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=r_ScopedGlideSystemDebug_String_Object_Object_Object)
(must be enabled)

All of the logging methods write to the System Log. Pass strings, variables that resolve to strings, or methods that resolve to strings into the logging methods.

```
gs.error("The value of the Short description field is " +  
current.short_description);
```

To view the log messages, open **System Logs > System Log > Application Logs** from the Application Navigator in the main ServiceNow browser window (not Studio).



The example script uses the *info()*, *warn()*, *error()*, and *debug()* methods.

Script

1

(function executeRule(current, previous /*null when async*/) {

2

3

// Scoped GlideSystem logging methods

4

gs.info("This is an info message from the NeedIt App.");

5

gs.warn("This is a warning message from the NeedIt App.");

6

gs.error("This is an error message from the NeedIt App.");

7

gs.debug("This is a debug message from the NeedIt App.");

8

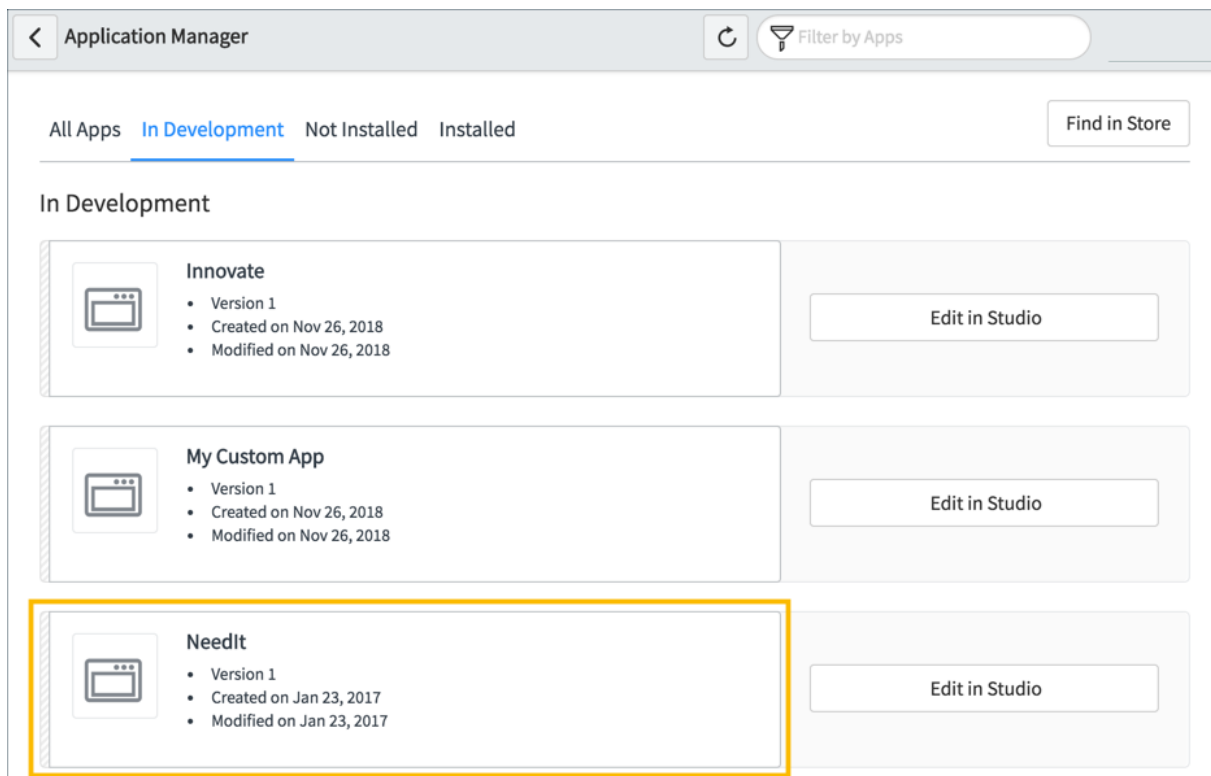
9

})(current, previous);

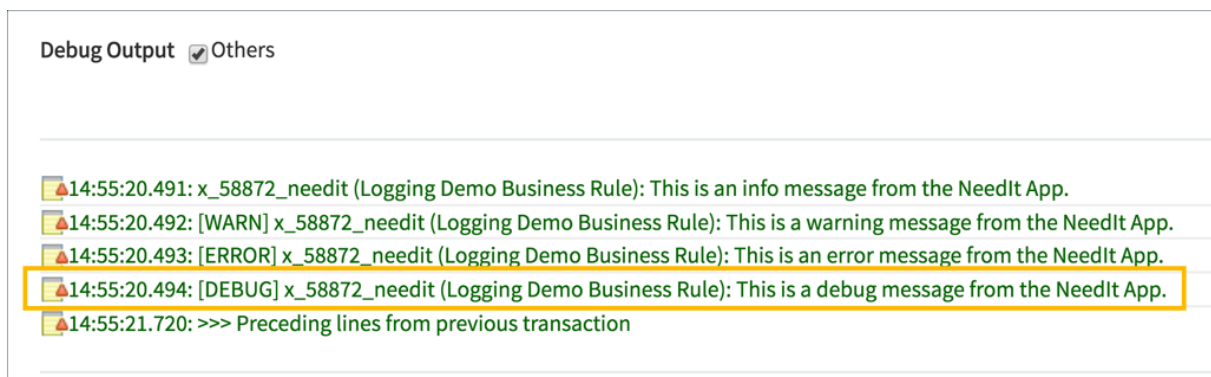
The log messages display the time created, verbosity level, message, scope, and script name.

App Log New Search Created ▾ Search						
All > Created on Today						
		Created ▾	Level	Message	App Scope	Source Script
<input type="checkbox"/>		2019-01-04 14:30:57	Information	This is an info message from the NeedIt App.	NeedIt	Business Rule: Logging Demo Business Rule
<input type="checkbox"/>		2019-01-04 14:30:57	Warning	This is a warning message from the NeedIt App.	NeedIt	Business Rule: Logging Demo Business Rule
<input type="checkbox"/>		2019-01-04 14:30:57	Error	This is an error message from the NeedIt App.	NeedIt	Business Rule: Logging Demo Business Rule

The scoped *GlideSystem* API has a *gs.debug()* method. By default, the scoped *GlideSystem debug()* method does not log information if the *gs.debug()* method is used in a script. To enable the *gs.debug()* method, use the Application Navigator to open **System Applications > My Company Applications**. Locate the application on one of the Application Manager's tabs. Click the application's card to open the application record (Do not click the **Edit in Studio** button for the application as that opens the application for editing in Studio.) In the Related Links section, click the **Enable Session Debug** link.



Debug information is written to the bottom of forms instead of the system log.



Debug Business Rule (Details)

To debug a *Condition* field script, enable detailed Business Rule Debugging. To enable debugging, use the Application Navigator to open **System Diagnostics >Session Debug > Debug Business Rule (Details)**. Open the form for a record of interest and force the Business Rule being debugged to execute by doing whatever is necessary to trigger the Business Rule. In the example, the *Logging Demo* Business Rule did not execute because the *Condition* field script did not evaluate to *true*.

Debug Output ☒ Business Rules ☐ Others

Apps ☐ Global ☐ Human Resources: Core ☒ NeedIt

15:00:03.942: Execute before update business rules on x_58872_needit_needit:NI002002 before engines (order <1000)

15:00:03.946: App:NeedIt === Skipping 'Logging Demo Business Rule' on x_58872_needit_needit:NI002002; condition not satisfied: Filter Condition: state=13^priority=1^EQ

15:00:03.947: Finished executing before update business rules on x_58872_needit_needit:NI002002 before engines (order <1000)

To turn off detailed Business Rule debugging, navigate to **System Diagnostics > Session Debug > Disable All**.