

Scripting for Non-Scripters

Rocky Mountain Netsuite User Group

Table of Contents

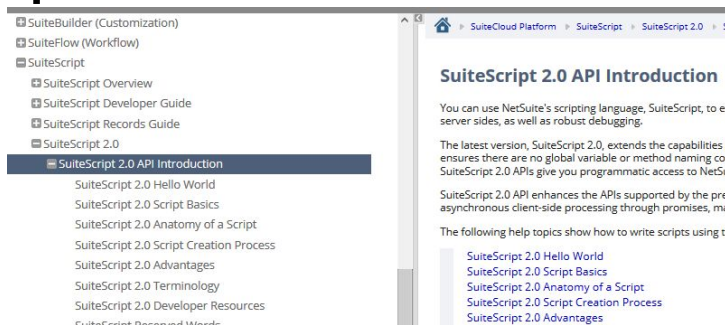
- What is SuiteScript?
- Why use SuiteScript?
- What types of scripts are there? / When to use each type?
- Where are Scripts found?
- What's Running?
- Script Permissions
- Script Logging
- Noteworthy Related Items
- Script Caveats and Limitations
- Learning More

```
257 function summarize(summary)_
258 {
259     var emailBody = "";
260     var scriptParams = getScriptParams();
261
262     emailBody += "<h2>Script Parameters</h2>";
263     emailBody += "<table>";
264     emailBody += "<tbody>";
265     for(var key in scriptParams) {
266         var value = scriptParams[key];
267         emailBody += "<tr><td>" + key + "</td><td>: " + value + "</td>
</tr>\n";
268     }
269     emailBody += "</tbody>";
270     emailBody += "</table>";
271
272     emailBody += "If needed, more Detailed Logging Exists in the Script Logs,
specifically in the 'Details' result column. Run a Netsuite 'Server Script Log'
search.";
273
274
275
276     try {
277         var emailAttachment = null;
278         if(scriptParams.csvFileID) {
279             emailAttachment = file.load({id:scriptParams.csvFileID});
280         }
281         vlib.sendMapReduceSummaryEmail(summary, "Mass Credit/Rebill Script
Complete", emailBody, scriptParams.ccList, emailAttachment);
282     } catch(e) {
283         log.error({title:"Exception Sending Summary Email:" + e.message,
details:[e.message, e.stack]});
```

CPE Credits? Check-In Keyword

FALL

What is SuiteScript?



JavaScript API within Netsuite which offers extensive customization abilities.

- An API built on top of the JavaScript language.
 - SuiteScript 1.0, 2.0, 2.1
- Runs on Netsuite “server” and/or user’s “client” device.
- Able to be scheduled, respond to events, and create custom “API’s” for third party integrations.

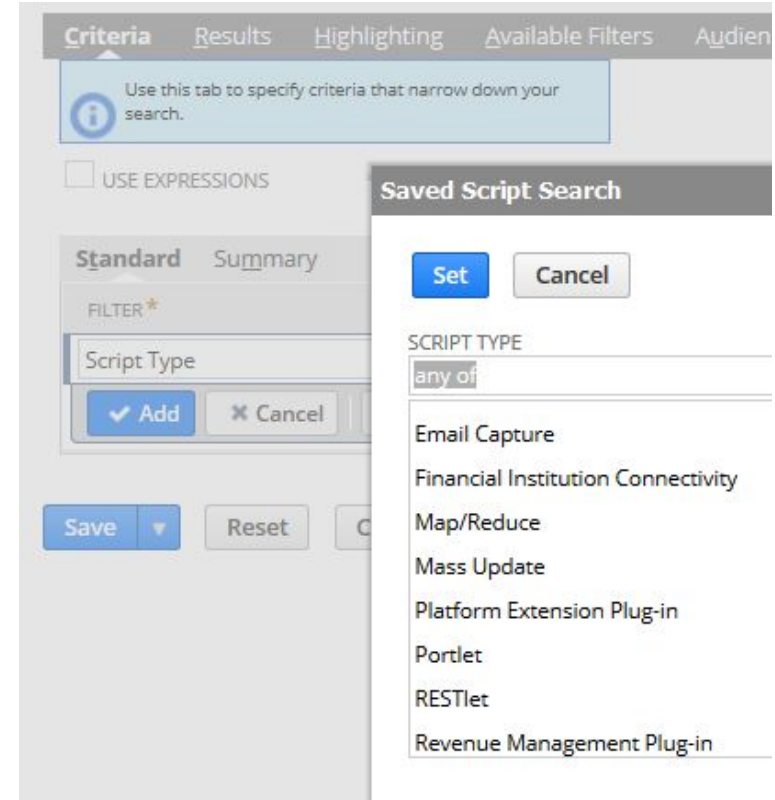
Why use SuiteScript?

Here are some examples of what can be done with SuiteScript.

- Enable functionality found nowhere else within Netsuite.
- Validating Data.
- Custom sublists on forms.
- Mass delete / update.
- PDF template magic.
- Perform complex operations on each row of a search result.
- SuiteQL queries
- Interact with a non-netsuite system via API.

What “types” of Scripts? ...

- Suitelet
- UserEvent
- Client
- MapReduce
- Workflow Action
- Others ...
 - Portlet
 - Restlet
 - Mass Update
 - Scheduled
 - SDF Install, Bundle Install



Types of Scripts - Suitelets

Used to create Netsuite-looking User Interfaces.

- Custom UI which interfaces with other scripts behind the scenes.
- Custom UI which then interacts with 3rd party software behind the scenes.
- “Custom replacement” for existing Netsuite screens.
- Often used to query/manipulate custom records.

The screenshot displays the Oracle NetSuite user interface. At the top, the header includes 'ORACLE NETSUITE', a 'SANDBOX' environment indicator, and the 'Vertafore' logo. A search bar is located in the top right corner. Below the header is a navigation menu with icons and labels for 'Activities', 'Payments', 'Transactions', 'Lists', 'Reports', 'Analytics', 'Documents', and 'Setup'. The main content area is titled 'BillingPlatform - Import Transactions'. On the left, a 'STEPS' sidebar lists two steps: '1 Set Script Parameters' (which is highlighted) and '2 Confirm then Start Job'. The main panel for step 1 is titled 'Set Script Parameters' and contains the following text: 'With this script, you can manually 'kick off' the import of transactions, from BP's servers, in ultimately equate to becoming a single invoice line item.' Below this text are several input fields: 'BP URL:' with the value 'https://my.billingplatform.com/vertafore/', 'BP USERNAME:' with a text input field, 'MINIMUM COVERAGE START DATE:' with a date input field, 'MAXIMUM COVERAGE END DATE:' with a date input field, and '(OPTIONAL) WHO TO EMAIL WHEN JOB IS DONE:' with a text input field. At the bottom of the panel, there is a link for 'BILLINGPLATFORM CONFIGURATION' and 'BillingPlatform Configuration Settings'.

Types of Scripts - UserEvent

Most useful type. Used to respond to changes in record data. Also useful for printing calculated data.

- Displaying a complex calculated value.
- Displaying “custom page” fields. (eg: Ad-hoc ‘custbody’ fields / sublists)
- “Before/After Save” data validations and actions.
- PDF template magic.
- Can run regardless of “context” (CSV, UI, Scripts, etc.)

```
87 * function validateItemFields (scriptContext) { // from v4ue1_CreditMemo
88 *   if (record.Type.CREDIT_MEMO != scriptContext.newRecord.Type) return;
89 *   var tRecord = scriptContext.newRecord;
90 *   var itemCount = tRecord.getLineCount({ 'sublistId' : 'item' });
91 *   for (var index = 0; itemCount >= index < itemCount; index++) {
92 *     var start = tRecord.getSublistValue({
93 *       'sublistId' : 'item',
94 *       'fieldId' : 'custool_revrec_start_date',
95 *       'line' : index
96 *     });
97 *     var end = tRecord.getSublistValue({
98 *       'sublistId' : 'item',
99 *       'fieldId' : 'custool_revrec_end_date',
100 *       'line' : index
101 *     });
102 *     if (start != undefined
103 *         && end != undefined) continue;
104 *     // else exception when start or end is undefined
105 *     throw error.create({
106 *       'name' : 'NOT_PERMITTED',
107 *       'message' : 'Please enter VF Rev Rec Start Date and VF Rev Rec End Date for
each line'
});
108 *   }
109 *   log.debug({
110 *     'title' : 'beforeSubmit.validateItemFields(...)',
111 *     'details' : 'Rev. Rec. Dates set before submit'
112 *   });
113 * }
114 *
```


Types of Scripts - Client

Primarily used for “real-time” changes on the screen as the user interacts with a form. This script ONLY runs in the context of a user interfacing with a transaction form.

- Dynamically add/enable/disable or populate fields as the user enters data.
- Real-Time “alerts” for missing or incorrect data on a form.
- Can be used in combination with Suitelets

Import Assistant

STEPS

- 1 Scan & Upload CSV File
- 2 Import Options
- 3 File Mapping
- 4 Field Mapping
- 5 Save mapping & Start Import

Scan & Upload CSV File

IMPORT TYPE
Accounting
Choose the category of data to import.

RECORD TYPE
Chart of Accounts
Choose the record type of data to import.

CHARACTER ENCODING
Western (Windows 1252)
Choose another character encoding format if you use an international or Macintosh version of Microsoft Excel, or

CSV COLUMN DELIMITER
Comma
Select the symbol to be used as a column separator in the CSV files you import. This setting overrides the column

CSV File(s)

☒ ONE FILE TO UPLOAD
☐ MULTIPLE FILES TO UPLOAD

Select...

These values are dependent upon "Import Type" selected

Types of Scripts - Map/Reduce

Best for “repetitive” operations. These can be scheduled, and are ideal for running the same logic on every row from a search result. Map/Reduce are also well suited for operating on very large data sets.

- DiY Invoicing, Dunning, Payment Generation
- Mass Create, Updates or Deletes, even if complex logic is needed.
- Creating CSV files from complex searches.
- Runs in Parallel using “SuiteCloud” processors!
- Tip: Use these instead of “Scheduled” Scripts.

```
29 * }
30 * Results {
31 *   Internal ID - GROUP
32 * }
33 */
34 function getInputData() {
35
36   log.audit({title:"Start MassApprove Map/Reduce Script"});
37   var scriptParams = getScriptParams();
38   return search.load({id:scriptParams.searchID});
39 }
40
41 function map(context) {
42
43   var searchResults = JSON.parse(context.value);
44
45   try {
46     var params = new Object;
47     params.id = searchResults.values["GROUP(internalid)"].value;
48     params.type = record.Type.JOURNAL_ENTRY;
49     var thisJournal = record.load(params);
50   } catch(e) {
```

Types of Scripts - Workflow Action

Workflows are great, but have known limitations. These scripts can supplement existing Workflows.

- Called from within a Workflow.
- Great for handling sublist data.
- Can reference records other than the record you are currently operating on.
- “Consolidating” many Workflow actions into a single action.

The screenshot displays the Oracle NetSuite interface with the 'Workflow: Requisition - Go-live Approval WF' configuration page. The left pane shows a workflow diagram with steps: START, Requisition Initiation, Finance Approval, CFO Approval, and Create Purchase Order. The right pane shows the 'Workflow Action' configuration for the 'Create Purchase Order' step.

Workflow Action Configuration:

- Basic Information:**
 - WORKFLOW: Requisition - Go-live Approval WF
 - STATE: Create Purchase Order
 - TYPE: Custom Action: VTF-WF1 Auto Create PO from Requisition
 - ID: workflowaction2221
 - TRIGGER ON: Entry
 - EVENT TYPES:
 - CONTEXTS:
 - ☐ INACTIVE
- Parameters:**
 - STORE RESULT IN: Created PO # (State)

Types of Scripts - “Other” Script Types

Worth noting, but you’ll probably not use these.

- **Scheduled** - Do “something” on a schedule.
- **Mass Update** - Complex operations on each row of a search result.
- **Portlet** - Create portlets based on custom and/or complex data.
- **Restlet** - Create “REST” web services which external systems can access.
- **SDF Install, Bundle Install** - Used by Developers.

Mass Updates	
ACTION	
+	Files
+	Group Records
+	Customer Support and Service
+	Transactions
+	Reports
+	Items
+	Amortization Schedules
+	Individuals
+	Roles & Permissions
+	Templates
+	Customer Web Site Assignment
+	Item Collection
-	Custom Updates
	BillingPlatform Transactions
	BillingPlatform - Mass Delete BPTxn
	Contact
	Last Sales Activity Mass Update
	Customer
	Customer Payment
	MassDelete Customer Payment Records
	Unapply Unapproved Customer Payments

Where are Scripts Found? ...

Scripts have three “parts”. There is a text file with “code” in it, a “Script” Record, and a “Script Deployment” record.

- File Cabinet - “SuiteScripts” folder
- Page: Scripts
- Page: Script Deployments

File

MassDeleteJournalEntry_mu2.js

Save

Cancel

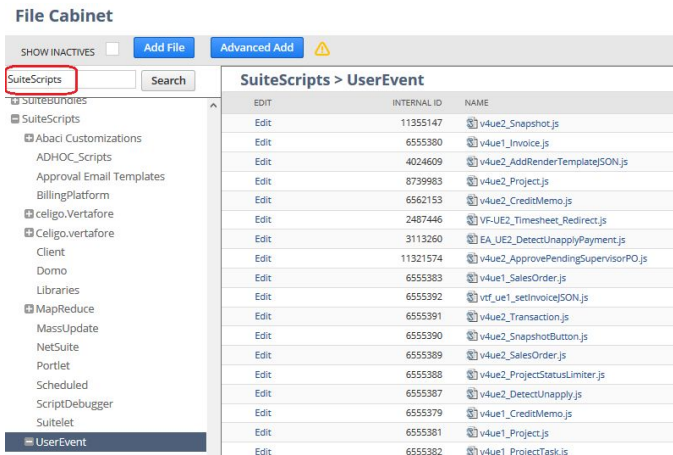
CONTENT

```
1 * /**
2  * @NApiVersion 2.x
3  * @NScriptType MassUpdateScript
4  * @NModuleScope SameAccount
5  */
6 define(['/SuiteScripts/VF 2.0 Scripts/VFDelete_lib'],
7 function(vfdelete) {
8
9     function each(params) {
10         try {
11             if (vfdelete.canDeleteJournalEntry(params)) {
12                 vfdelete.deleteJournalEntry(params);
13             } else {
14                 log.audit({title:"JournalEntry can NOT be deleted
15 details:params});
16             }
17         } catch(e) {
18             log.audit({title:"canDeleteJournalEntry() exception:
19 details:[e.message, e.stack, params]});
20         }
21     }
22 }
```

Where are Scripts? - File Cabinet

Just another file in the file cabinet, with some caveats. These files are referred later on as “source code.”

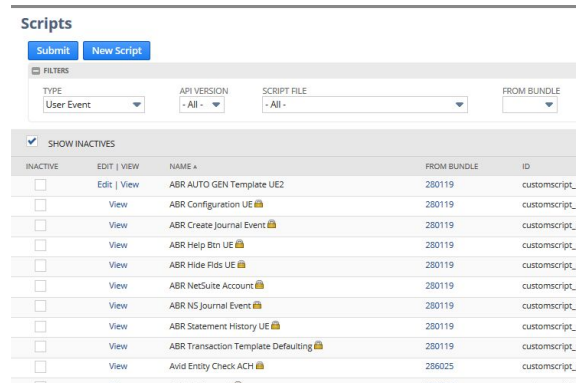
- Scripts MUST be in the “SuiteScripts” folder, or any subfolder thereof.
- Scripts must end with “.js”
- Script files are parsed for syntax errors upon upload.
- Files can be edited from within File Cabinet.



Where are Scripts? - Page: Scripts

A netsuite record which defines what the script should do and where the source code is.

- Tells Netsuite that source code is “executable.”
- Gives the script a plain english name.
- Formal place to define “input parameters” for the script.
- View “script logs” for times this specific script ran. (24 hour window)
- Script records can be searched.



The screenshot shows the Netsuite 'Scripts' page. At the top, there are 'Submit' and 'New Script' buttons. Below them is a 'FILTERS' section with dropdown menus for 'TYPE' (set to 'User Event'), 'API VERSION' (set to '- All -'), 'SCRIPT FILE' (set to '- All -'), and 'FROM BUNDLE'. A 'SHOW INACTIVES' checkbox is checked. The main table lists various scripts with columns for 'INACTIVE', 'EDIT | VIEW', 'NAME', 'FROM BUNDLE', and 'ID'. The scripts listed include 'ABR AUTO GEN Template UE2', 'ABR Configuration UE', 'ABR Create Journal Event', 'ABR Help Btn UE', 'ABR Hide Fids UE', 'ABR NetSuite Account', 'ABR NS Journal Event', 'ABR Statement History UE', 'ABR Transaction Template Defaulting', and 'Add Entity Check ACH'.

INACTIVE	EDIT VIEW	NAME	FROM BUNDLE	ID
<input type="checkbox"/>	Edit View	ABR AUTO GEN Template UE2	280119	customscript_r
<input type="checkbox"/>	View	ABR Configuration UE	280119	customscript_r
<input type="checkbox"/>	View	ABR Create Journal Event	280119	customscript_r
<input type="checkbox"/>	View	ABR Help Btn UE	280119	customscript_r
<input type="checkbox"/>	View	ABR Hide Fids UE	280119	customscript_r
<input type="checkbox"/>	View	ABR NetSuite Account	280119	customscript_r
<input type="checkbox"/>	View	ABR NS Journal Event	280119	customscript_r
<input type="checkbox"/>	View	ABR Statement History UE	280119	customscript_r
<input type="checkbox"/>	View	ABR Transaction Template Defaulting	280119	customscript_r
<input type="checkbox"/>	View	Add Entity Check ACH	286025	customscript_r

Where are Scripts? - Script Deployments

Defines a scenario where we would like a particular script record to actually run.

- “Script Deployments” subtab of a Script Record.
- Page: Script Deployments
- (If Avail) Define schedule on which the script will run.
- Defines which user the script will run as (permissions.)
- Define CPU utilization and prioritization.
- Define specific Script Parameters

The screenshot displays the 'Script Deployment' configuration page. At the top, there are buttons for 'Save', 'Cancel', 'Reset', 'Change ID', and 'Actions'. The main configuration area includes fields for 'SCRIPT' (ZAB Revenue Link Linking), 'TITLE' (ZAB Revenue Link Linking), and 'ID' (customdeployzab_revenue_link_linking_mr1). A 'DEPLOYED' checkbox is checked. On the right side, there are dropdown menus for 'STATUS' (Scheduled), 'LOG LEVEL' (Audit), 'EXECUTE AS ROLE' (Administrator), 'PRIORITY' (Standard), 'CONCURRENCY LIMIT' (25), 'YIELD AFTER MINUTES' (15), and 'BUFFER SIZE' (1). A 'SUBMIT ALL STAGES AT ONCE' checkbox is also checked. At the bottom, there is a 'Schedule' tab with radio buttons for 'SINGLE EVENT', 'DAILY EVENT' (selected), 'WEEKLY EVENT', 'MONTHLY EVENT', and 'YEARLY EVENT'. The 'DAILY EVENT' section has a 'Repeat every 1 day(s)' field. Below this, there are fields for 'START DATE' (1/30/2019), 'START TIME' (4:00 am), 'REPEAT' (every 8 hours), and 'END BY'. A 'NO END DATE' checkbox is checked.

Coffee Break!



What's Running? ...

How to tell if a script is currently running or, in the case of UserEvents, if a script will run when a record is accessed.

- Page: Map/Reduce Script Status
- Page: Scripted Records
- Page: Mass Updates
- How to Enable/Disable Scripts

What's Running? - Map/Reduce

Page: Map/Reduce Script Status

- Shows actively running scripts, and current “stage”
- Allows cancelling “queued up / not yet started” scripts.
- CANNOT stop an already started script.

Map/Reduce Script Status

[Refresh](#)

FILTERS

DATE: today FROM: 8/23/2020 TO: 8/23/2020 SCRIPT: - All -

DEPLOYMENT ID: - All - STATUS: - All -

☒ SHOW INACTIVES 8/23/2020(1) TOTAL: 139

SCRIPT	DEPLOYMENT ID	DATE CREATED	START DATE	END DATE	STATUS	STAGE	PRIORITY	CONCURRENCY	DETAILS	CANCEL
V4 Add ZAB Link onto Project	customdeploy_v4mr2_zablinkonprj_sched	8/23/2020 4:01:43 pm	8/23/2020 4:01:46 pm	8/23/2020 4:04:44 pm	Complete		Standard	2	Details	
V4 Mass Render Snapshots	customdeploy_v4mr2_modified_payments	8/23/2020 4:01:43 pm	8/23/2020 4:01:45 pm	8/23/2020 4:02:42 pm	Complete		Standard	15	Details	
V4 Mass Render Snapshots	customdeploy_v4mr2_modified_invoices	8/23/2020 4:01:42 pm	8/23/2020 4:01:45 pm	8/23/2020 4:02:49 pm	Complete		Standard	15	Details	
ZAB Automation MR	customdeployzab_automation_mr_sch	8/23/2020 4:01:40 pm	8/23/2020 4:01:45 pm	8/23/2020 4:02:22 pm	Complete		Standard	25	Details	
V4 Mass Render Snapshots	customdeploy_v4mr2_modified_credits	8/23/2020 4:01:40 pm	8/23/2020 4:01:42 pm	8/23/2020 4:02:39 pm	Complete		Standard	15	Details	

What's Running? - Scripted Records

Scripted Records				
FILTERS				
SCRIPT - All -				
<input type="checkbox"/> SHOW UNDEPLOYED				
Lease — Zab Plan < > TOTAL				
EDIT VIEW	RECORD *	SCRIPT ID	SCRIPTS	WORKFLOWS
Edit View	Phone Call	phonecall	1	0
Edit View	Project	job	21	14
Edit View	Project Task	projecttask	12	3
Edit View	Prospect	prospect	3	1
Edit View	Purchase Order	purchaseorder	11	8
Edit View	Requisition	purchase requisition	5	6
Edit View	Resource Allocation	resourceallocation	9	3
Edit View	Revenue Arrangement	revenuearrangement	1	0

Page: Scripted Records

- Viewed on a “per record type” basis.
- UserEvent and Client Scripts
- Shows which scripts will trigger, and on which events.
- Shows the SEQUENCE in which scripts will fire.

What's Running? - Mass Updates

Page: Mass Updates

- Link pointing to the last Mass Update you ran.
- Tip: Your custom scripts will show under the “Custom Updates” section



What's Running? - How to Enable/Disable Scripts

- Deployments vs Script
- “Inactive” vs “Deployed”
- “Testing” Status

Script Deployment

[Save](#) [Cancel](#) [Reset](#) [Save and Debug](#) [Change ID](#) [Actions](#)

SCRIPT
[AS NEEDED]-MR2 DeleteRecords By SSearch

TITLE *
[AS NEEDED]-MR2 DeleteRecords By SSearch

ID
customdeploy1

☐ DEPLOYED

STATUS *
Testing

SEE INSTANCES
[Status Page](#)

LOG LEVEL
Debug

EXECUTE AS ROLE
Administrator

Script

[Save](#) [Cancel](#) [Reset](#) [Change ID](#) [Actions](#)

TYPE
Map/Reduce

NAME *
[AS NEEDED]-MR2 DeleteRecords By SSearch

ID
customscript619

API VERSION
2.0

DESCRIPTION

OWNER
Joe Son

☐ INACTIVE

[Scripts](#) [Parameters](#) [Unhandled Errors](#) [Execution Log](#) [Deployments](#) [System Ne](#)

TITLE	ID	DEPLOYED	STATUS	LOG LEVEL	REC
[AS NEEDED]-MR2 DeleteRecords By SSearch	customdeploy1		Testing	Debug	one 10/9

Script Permissions

- “Run As Administrator”
- Status = “Testing”
- System Information Fingerprints
- User Events being triggered from Scheduled or Map/Reduce

Script Deployment Map/Reduce Script Deployment

[Edit](#) [Back](#) | [Actions](#) ▾

SCRIPT
V4 Snapshot Transactions

TITLE
V4 Snapshot Invoices Modified 1 Day Ago

ID
customdeploy_v4mr2_snapshot_invoices_00

☒ DEPLOYED

STATUS
Not Scheduled

[SEE INSTANCES](#)

[Status Page](#)

LOG LEVEL
Debug

**EXECUTE AS ROLE
Administrator**

PRIORITY
Standard

CONCURRENCY LIMIT
20

☒ SUBMIT ALL STAGES /

Script Deployment User Event Script Deployment

[Save](#) [Cancel](#) [Reset](#) [Change ID](#) | [Actions](#) ▾

SCRIPT
V4 Project UserEvent

APPLIES TO *
Project

ID
customdeploy_v4ue1_project

☒ DEPLOYED

STATUS *

Released

Testing

Released

Debug

EXECUTE AS ROLE
Current Role

Audience • **Scripts** • **Parameters** • **Context Filtering** • **Execution Log** • **System Notes**

ROLES ☒ Select All

SUBSIDIARIES

EMPLOYEES ☐ Select All

Script Logging ...

Developers can write code which “logs” some event happening. This is useful for tracing how a script ran, and is also used to give indication if the script experienced problems and why.

- Page: Scripts | “Execution Logs” subtab
- Searching “Server Script Logs”
- Logs do not persist for very long
- Favorite Searches ...

Custom Server Script Log Search 2: Results

[Return To Criteria](#)

FILTERS

STYLE
Normal

#	INTERNAL ID ▾	VIEW	TITLE	TYPE	DATE	TIME
1	6887744	View	final check JSON	Debug	8/23/2020	5:08 am
2	6887743	View	timeTaskToAdd	Debug	8/23/2020	5:08 am
3	6887742	View	mtjson	Debug	8/23/2020	5:08 am
4	6887741	View	itemType // itemSku	Debug	8/23/2020	5:08 am
5	6887740	View	project name	Debug	8/23/2020	5:08 am
6	6887739	View	running project queue	Debug	8/23/2020	5:08 am
7	6887738	View	size of queue	Audit	8/23/2020	5:08 am
8	6887737	View	Project Queue to Process // Last Proc ID	Audit	8/23/2020	5:08 am
9	6887736	View	final check JSON	Debug	8/23/2020	5:01 am
10	6887735	View	timeTaskToAdd	Debug	8/23/2020	5:01 am
11	6887734	View	mtjson	Debug	8/23/2020	5:01 am
12	6887733	View	itemType // itemSku	Debug	8/23/2020	5:01 am
13	6887732	View	project name	Debug	8/23/2020	5:01 am

Script Logging - “Execution Logs” subtab

- Used often by developers
- See logs for all “deployments” of this script.
- Also visible from the “Script Deployment” record (only logs for that deployment)
- Only a 24 hour window of logs are displayed.


The screenshot displays the Salesforce Scripting tool interface. At the top, the 'Script' section includes buttons for 'Edit', 'Back', 'Deploy Script', and an 'Actions' dropdown. Below this, script details are listed: TYPE (Map/Reduce), NAME (V4 Mass Render Snapshots), ID (customscript_v4mr2_massrendersnapshots), and API VERSION (2.0). A horizontal tab bar contains 'Scripts', 'Parameters', 'Unhandled Errors', 'Execution Log' (highlighted with a red box), 'Deployments', and 'System Notes'. Below the tabs, there are dropdowns for 'VIEW' (Default Script Notes View) and 'TYPE' (- All -). A row of buttons includes 'Customize View', 'Remove all', and 'Refresh'. The main area features a table with three columns: #, VIEW, and TYPE. The table contains three log entries, each with a 'View' link, a 'Debug' type, a title, a date, a time, a user, and a details link.

#	VIEW	TYPE	TITLE	DATE	TIME	USER	DETAILS
1	View	Debug	Email customdeploy_v4mr2_modified_invoices	8/23/2020	4:02 pm	-System-	({"subject": "cus Invoice-", "sear Invoices20200i ("type": "mapre
2	View	Debug	Email customdeploy_v4mr2_modified_payments	8/23/2020	4:02 pm	-System-	({"subject": "cus Payments-", "se Payments2020 ("type": "mapre
3	View	Debug	Email customdeploy_v4mr2_modified_credits	8/23/2020	4:02 pm	-System-	({"subject": "cus CreditMemo-",

Script Logging - Saved Searches

Script “logs” are saved for a few days and can be searched via saved search. This can give a good history of which scripts ran, and when.

- Can be Emailed, setup as Portlet Dashboard
- Favorite Script Searches ...

New Saved Search	
Role	
Save SS Data To File Cabinet	
Saved Search	
Scheduled Script Instance	
Script	
Script Deployment	
Sent Email List	
Server Script Log	
Shipment Package	

Favorite Logging Search: Recent Script Errors

Most developers use log files to “follow” how their script is doing and log cases where the script experienced an error. These errors can easily be lost. Here, we show a way to surface up script errors.

- Type: Server Script Logs
- Criteria: “Type=Error”
- Results: Sort by date, desc
- Results can go back approximately 30 days.

Saved Server Script Log Search

Recent Script Errors (Detail)

Save & Run ▼

Reset

Cancel

Preview

New Template

Change ID

SEARCH TITLE ★
Recent Script Errors (Detail)

ID
customsearch8160

OWNER
Mike Kachline ▼

☒ PUBLIC

☐ AVAILABLE AS LIST VIEW

CriteriaResultsHighlightingAvailable FiltersAudienceRolesEmailAudit Trail

Use this tab to specify criteria that narrow down your search.

☐ USE EXPRESSIONS

Standard • Summary

FILTER ★

DESCRIPTION ★

Typeis Error

▼

▼ Add

✕ Cancel

+ Insert

🗑 Remove

Favorite Search: Recently Changed Scripts

The screenshot shows the Netsuite search criteria configuration interface. At the top, there is a navigation bar with tabs: Criteria, Results, Highlighting, Available Filters, Audience, Roles, Email, Audit Trail, Execution Log, and Search. Below the navigation bar, there is a blue information box with a question mark icon and the text: "Use this tab to specify criteria that narrow down your search." Below the information box, there is a checkbox labeled "USE EXPRESSIONS" which is currently unchecked. Below the checkbox, there is a table with two tabs: "Standard" (selected) and "Summary". The table has three columns: FILTER, DESCRIPTION, and FORMULA. The table contains the following rows:

FILTER	DESCRIPTION	FORMULA
Date	is on today	
Type	is Change	
Field	is File Content Hash	
Formula (Text)	contains .js	{record}
Formula (Numeric)	is greater than 0	REGEXP_INSTR({record}, '.js\$')

Below the table, there is a dropdown menu with a downward arrow. At the bottom of the interface, there are four buttons: "Add" (with a checkmark icon), "Cancel" (with an X icon), "Insert" (with a plus icon), and "Remove" (with a trash can icon).

Netsuite does not have a good “change control” process for scripts. However, this search trick is very useful in notifying you if any of your scripts have changed.

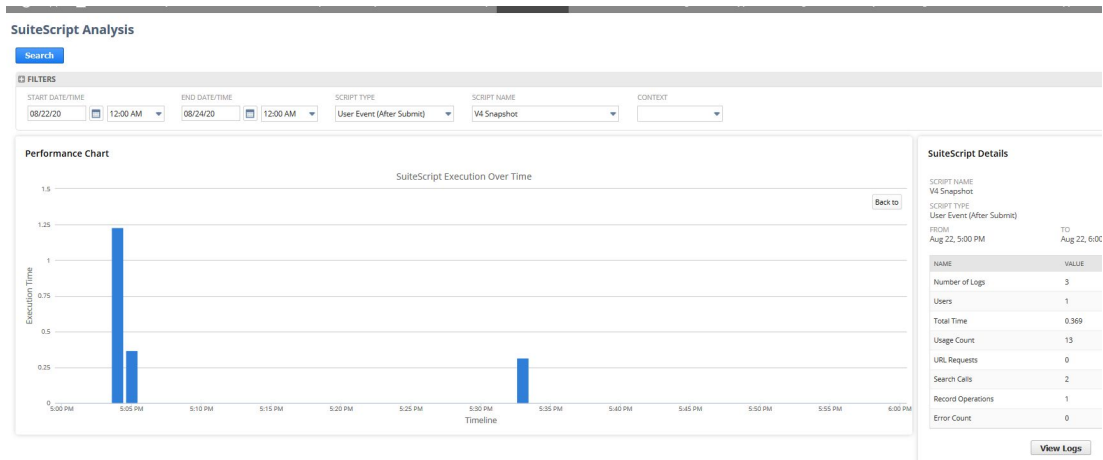
- “System Note” Search
- Criteria: “Field is ‘File Content Hash’” / Date is today.
- Tells which script files have changed in the past day.

Noteworthy Items ...

Other tools related to scripting which developers typically use, which you should know about.

- Script Performance Monitor
- Script Debugger
- SDF Projects

Noteworthy - Script Performance Monitor

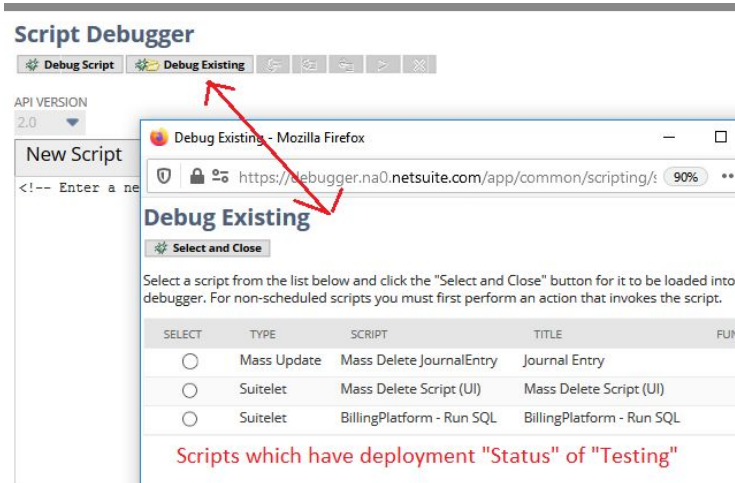


- Free Netsuite bundle ID=67350 “Application Performance Management”
- Identify which scripts are running slow, and why.

Noteworthy - Script Debugger

Useful developer tool which allows devs to “step through” code as it is running real-time.



- Page: Script Debugger
- Offers standard debugger functionality (step in/over, viewing of variables real-time.)
- Does NOT work with Map/Reduce scripts.
- Can be used for “Ad-Hoc” scripts.
- “Existing” scripts must be in “Testing” status.



Noteworthy - SDF Projects

A means of installing “in-house apps” onto your system, and associated audit trail.

- Page: Deployment Audit Trail
- Think of these as “Desktop deployable bundles.”
- Often integrated with code “IDE” tool (Webstorm, Eclipse)
- Requires additional permissions

Deployment Audit Trail									
<div>Refresh</div>									
<div> </div>									
PUBLISHER ID	APP ID	NAME	VERSION	ACTION	STATUS	START DATE	END DATE	USER	ERROR REFERENCE
		DeployNetSuite		INSTALL	COMPLETE	8/20/2020 10:43 am	8/20/2020 10:44 am	David Orozco	
		DeployNetSuite		INSTALL	FAILED_TO_VALIDATE	8/20/2020 10:41 am	8/20/2020 10:41 am	David Orozco	ke33d7opmbrei3c
		DeployNetSuite		INSTALL	COMPLETE	8/14/2020 10:30 am	8/14/2020 10:30 am	David Orozco	

Script Caveats and Limitations

- “Script Execution Limits”
- Sequencing of UserEvents
- Scripts calling Scripts
- Sequencing of Scripts vs Workflows
- Poor audit trail of changes
- Logging of Script activities
- SuiteScript 1.x code is very different from 2.x
- CSV import checkbox - “Run Server Suitescript”

Scripted Record

Edit

Back

NAME
Employee

Runs First

Runs Last

Script	OWNER	API VERSION	FROM BUNDLE
RSS UE Main	Ron Breaux	1.0	36477
Avid UE Entity	Eric Park	1.0	286025
AvidXchange Inactivated Employee	Account Provisioning	2.0	286025
Avid Entity Check ACH	Account Provisioning	2.0	286025
Generic Entity Initializer	Peter August	1.0	308852

Import Assistant

STEPS

1 Scan & Upload CSV File

2 Import Options

3 File Mapping

4 Field Mapping

5 Save mapping & Start Import

Import Options

Advanced Options

☐ LOG SYSTEM NOTES FOR CUSTOM FIELDS

Enable this option to create system notes during import of custom field data. Impacts performance; recommended only when custom fields require an audit trail.

☐ OVERWRITE MISSING FIELDS

For updates, enable this option to clear NetSuite fields mapped to CSV file fields that do not contain data.

☒ VALIDATE MANDATORY CUSTOM FIELDS

Enable this option to require mandatory custom field data to be present for records to be created.

☐ OVERWRITE SUBLISTS

For updates, enable this option to cause imported sublist data to completely replace existing sublist data, instead of selectively updating or being appended ... [more](#)

☒ IGNORE READ-ONLY FIELDS

This option, enabled by default, allows you to import CSV files containing values for read-only fields without causing errors.

☐ CUSTOM MULTI-SELECT VALUE DELIMITER

1

CSV DECIMAL DELIMITER

Period

Select the symbol to be used as a decimal in import. This setting overrides the decimal specified at Home > Set Preferences.

☐ USE MULTI-THREADING

If order does not matter for imported CSV to optimize performance. When multiple threads may not be submitted in order.

QUEUE NUMBER

1

Select an alternate queue (other than the default) to spread processing across multiple queues to improve performance.

☒ RUN SERVER SUITESCRIPT AND LOG

Check to specify that any server-side SuiteScript should be triggered for the current CSV import. Running server SuiteScript slows the save process.

Learning More

- Netsuite User Guide:
“SuiteScript Developer Guide”
- Netsuite Help “SuiteScript 2.0”
- Netsuite Professionals Slack
Forum: “SuiteScript” channel.

SuiteScript Developer Guide



Thank You! - Questions?

RMNSUG - Scripting for Non-Scripters

CPE Credits?

Checkout Keyword: TREES