Scripting for Non-Scripters

Rocky Mountain Netsuite User Group

Table of Contents

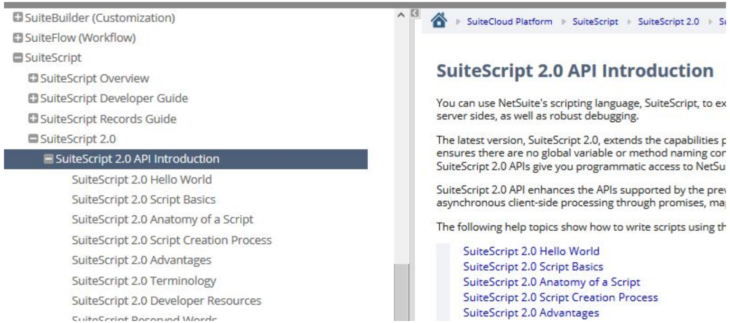
- What is SuiteScript?
- Why use SuiteScript?
- What types of scripts are there? / When to use each type?
- Where are Scripts found?
- What's Running?
- Script Permissions
- Script Logging
- Noteworthy Related Items
- Script Caveats and Limitations
- Learning More

```
function summarize(summary)
{
    var emailBody = "";
    var scriptParams = getScriptParams();

    emailBody += "<h2>Script Parameters</h2>";
    emailBody += "<table>";
    emailBody += "<tbody>";
    for(var key in scriptParams) {
        var value = scriptParams[key];
        emailBody += "<tr><td>" + key + "</td><td>: " + value + "</td>
</tr>\n";
    }
    emailBody += "</tbody>";
    emailBody += "</table>";

    emailBody += "If needed, more Detailed Logging Exists in the Script Logs,
specifically in the 'Details' result column. Run a Netsuite 'Server Script Log'
search.";



    try {
        var emailAttachment = null;
        if(scriptParams.csvFileID) {
            emailAttachment = file.load({id:scriptParams.csvFileID});
        }
        vflib.sendMapReduceSummaryEmail(summary, "Mass Credit/Rebill Script
Complete", emailBody, scriptParams.ccList, emailAttachment);
    } catch(e) {
        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.

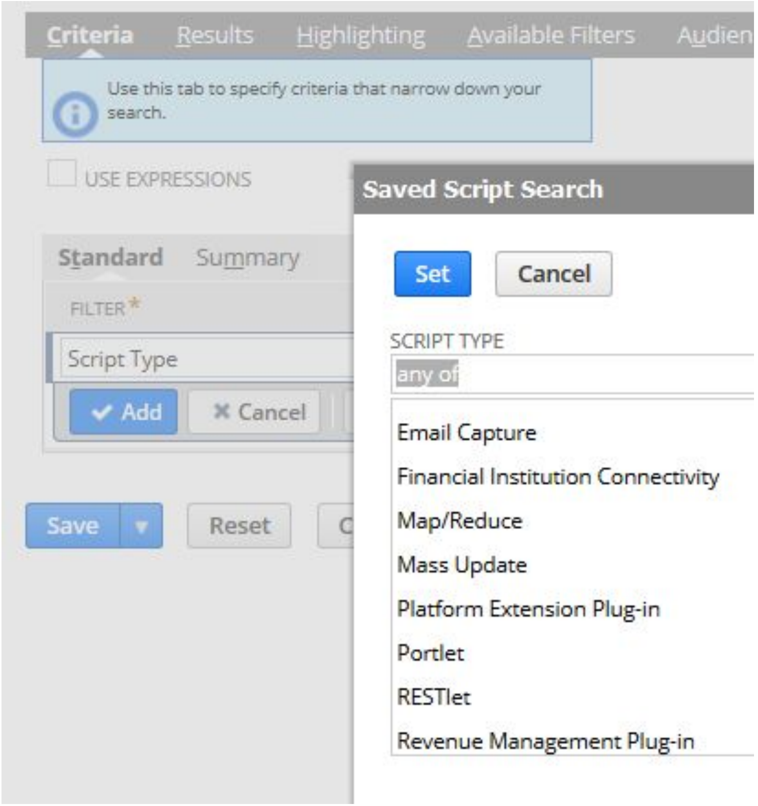
Why use SuiteScript?

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? ...

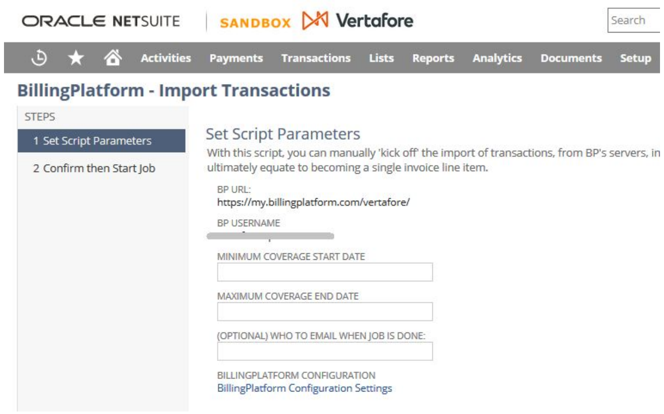
- Suitelet
- UserEvent
- Client
- MapReduce
- Workflow Action
- Others …
    - Portlet
    - Restlet
    - Mass Update
    - Scheduled
    - SDF Install, Bundle Install



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.



Types of Scripts - UserEvent

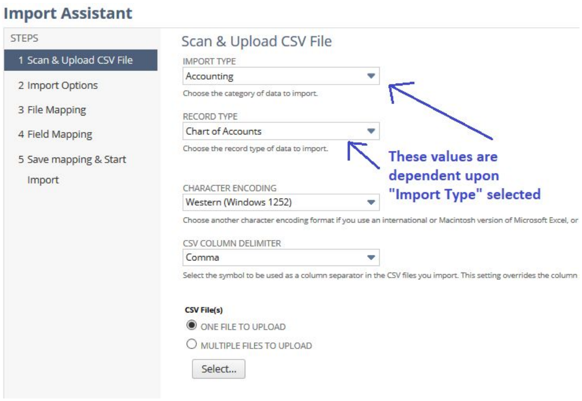
Most useful type. Used to respond to changes in record data. Also useful for printing calculated data.

- 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.)

```
function validateItemFields (scriptContext) { // from v4ue1_CreditMemo
    if (record.Type.CREDIT_MEMO != scriptContext.newRecord.type) return;
    var tRecord = scriptContext.newRecord;
    var itemCount = tRecord.getLineCount({ 'sublistId' : 'item' });
    for (var index = 0; itemCount && index < itemCount; index++) {
        var start = tRecord.getSublistValue({
            'sublistId' : 'item',
            'fieldId' : 'custcol_revrec_start_date',
            'line' : index
        });
        var end = tRecord.getSublistValue({
            'sublistId' : 'item',
            'fieldId' : 'custcol_revrec_end_date',
            'line' : index
        });
        if (start != undefined
            && end != undefined) continue;
        // else exception when start or end is undefined
        throw error.create({
            'name' : 'NOT_PERMITTED',
            'message' : 'Please enter VF Rev Rec Start Date and VF Rev Rec End Date for each line'
        });
    }
    log.debug({
        'title' : 'beforeSubmit.validateItemFields(...)',
        'details' : 'Rev. Rec. Dates set before submit'
    });
}
```

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

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.

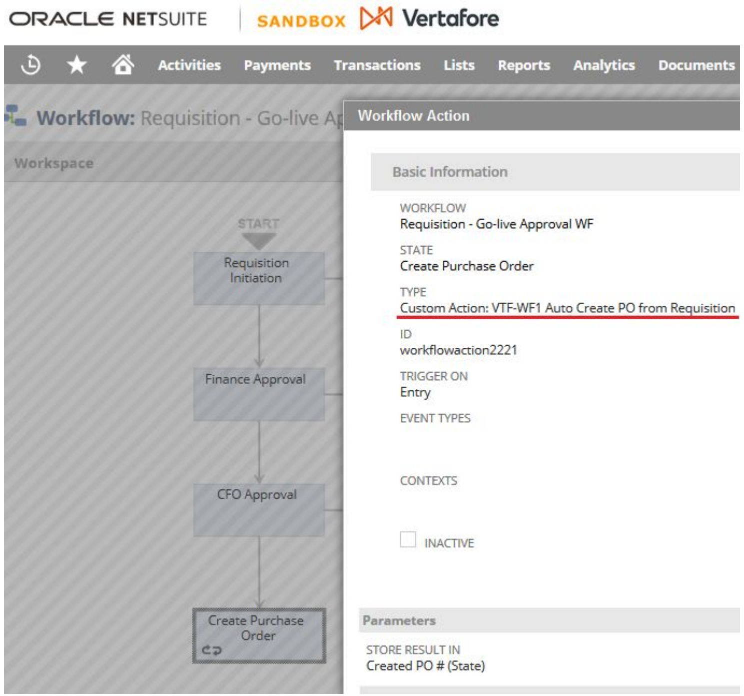
```
     * }
     * Results {
     *   Internal ID - GROUP
     * }
     */
    function getInputData() {

        log.audit({title:"Start MassApprove Map/Reduce Script"});
        var scriptParams = getScriptParams();
        return search.load({id:scriptParams.searchID});
    }

    function map(context) {

        var searchResults = JSON.parse(context.value);

        try {
            var params = new Object;
            params.id = searchResults.values["GROUP(internalid)"].value;
            params.type = record.Type.JOURNAL_ENTRY;
            var thisJournal = record.load(params);
        } catch(e) {
```

- 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.

Types of Scripts - Workflow Action

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

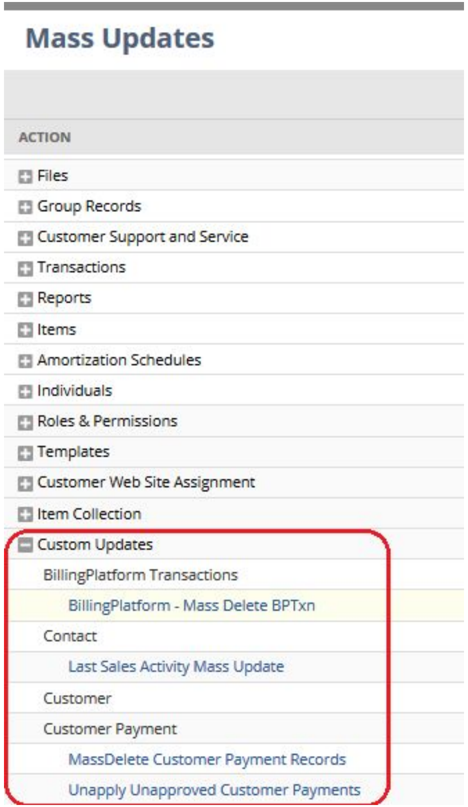
- Called from within a Workflow.
- Great for handling sublist data.
- Can reference records other than the record you are currently operating on.
- “Consolidating” many Workflow actions into a single action.



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.



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

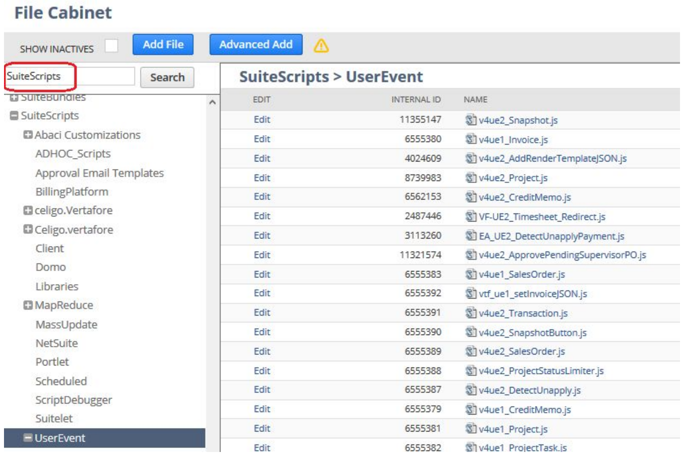
CONTENT

```
/**
 * @NApiVersion 2.x
 * @NScriptType MassUpdateScript
 * @NModuleScope SameAccount
 */
define(['/SuiteScripts/VF 2.0 Scripts/VFDelete_lib'],
function(vfdelete) {

    function each(params) {
        try {
            if (vfdelete.canDeleteJournalEntry(params)) {
                vfdelete.deleteJournalEntry(params);
            } else {
                log.audit({title:"JournalEntry can NOT be deleted
details:params});
            }
        } catch(e) {
            log.audit({title:"canDeleteJournalEntry() exception:
details:[e.message, e.stack, params]});
        }
    }

```

Where are Scripts? - File Cabinet

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

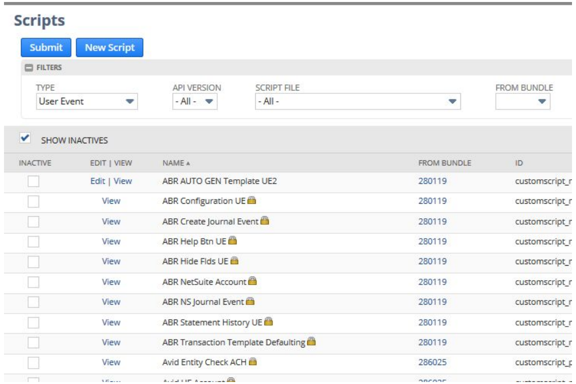
- Scripts MUST be in the “SuiteScripts” folder, or any subfolder thereof.
- Scripts must end with “.js”
- Script files are parsed for syntax errors upon upload.
- Files can be edited from within File Cabinet.



Where are Scripts? - Page: Scripts

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

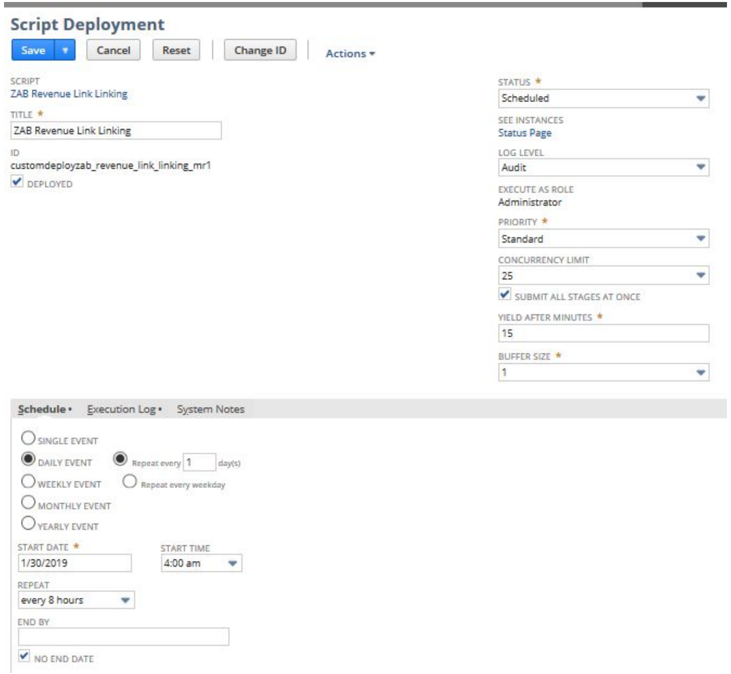
- Tells Netsuite that source code is “executable.”
- Gives the script a plain english name.
- Formal place to define “input parameters” for the script.
- View “script logs” for times this specific script ran. (24 hour window)
- Script records can be searched.



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



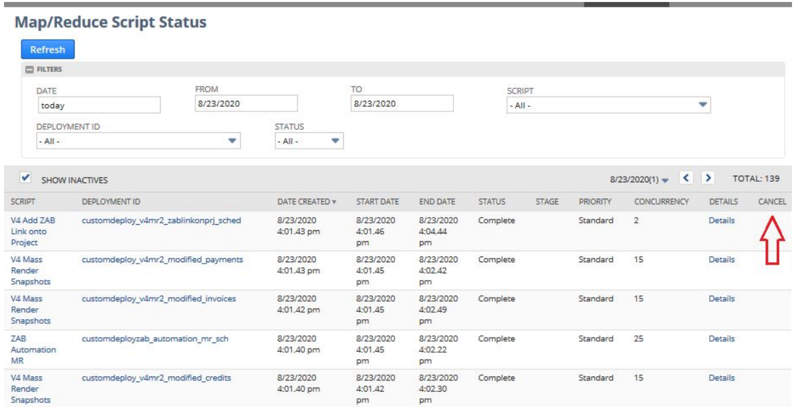
Coffee Break!

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


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 4:02.30 pm	Complete		Standard	15	Details	

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.

What’s Running? - Scripted Records

Scripted Records

FILTERS

SCRIPT

- All -

SHOW UNDEPLOYED

Lease — Zab Plan

TOTA

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	purchaserequisition	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.

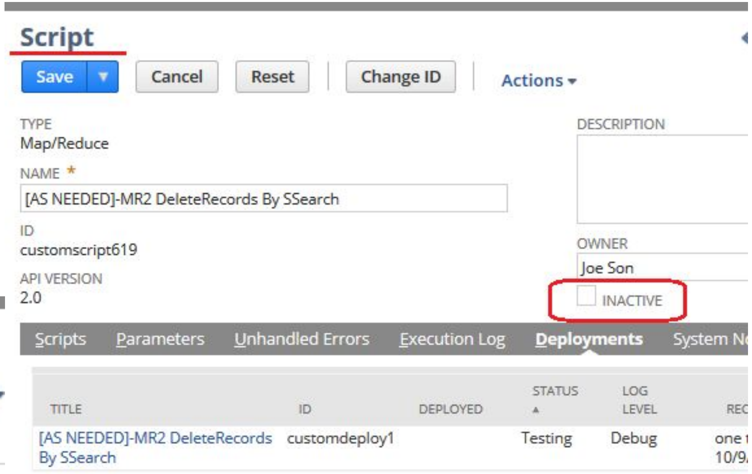
What’s Running? - Mass Updates

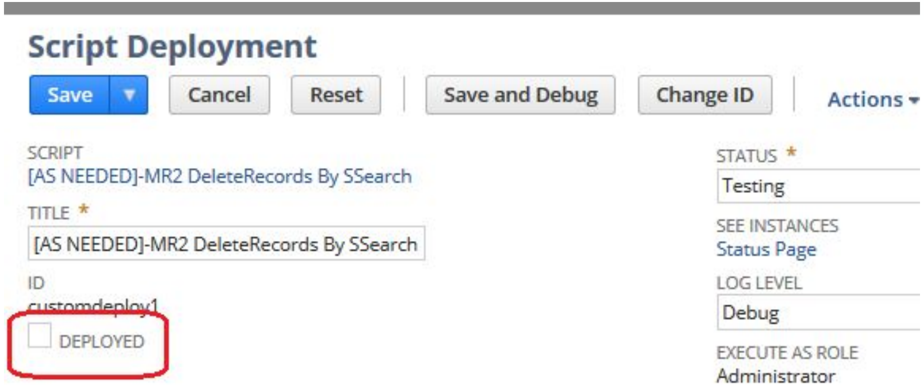
Page: Mass Updates

- 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

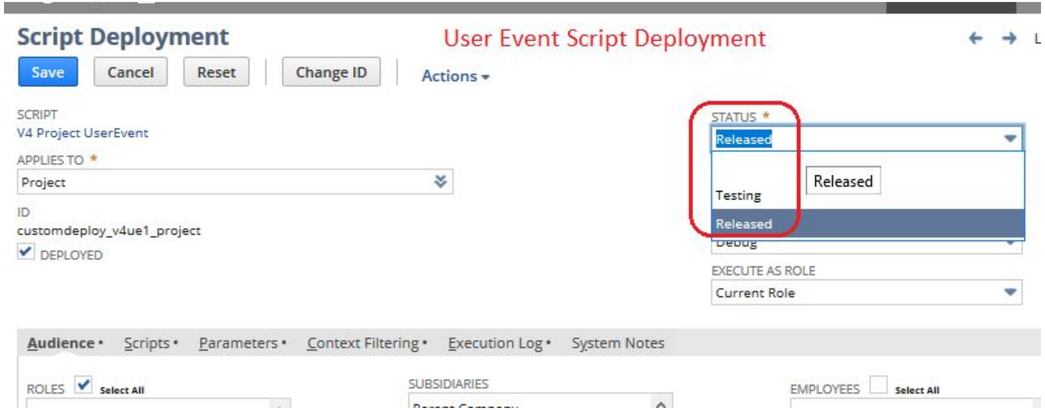
- Deployments vs Script
- “Inactive” vs “Deployed”
- “Testing” Status





Script Permissions

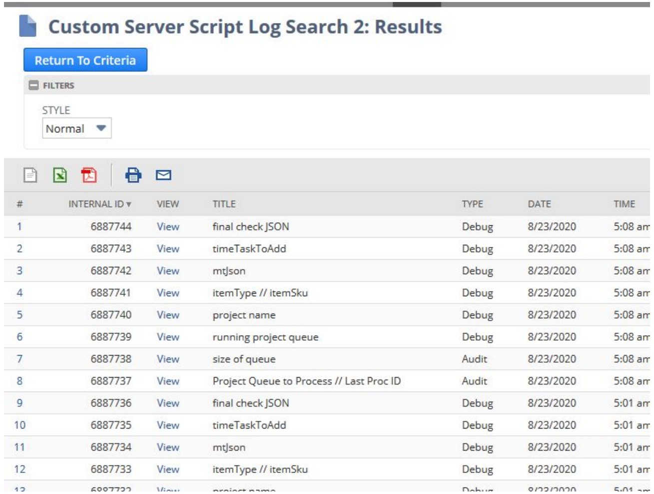
- “Run As Administrator”
- Status = “Testing”
- System Information Fingerprints
- User Events being triggered from Scheduled or Map/Reduce



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

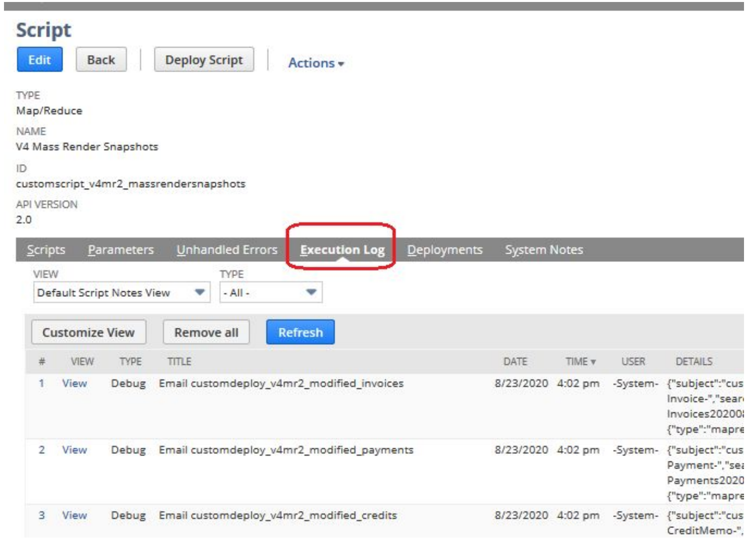
FILTERS

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

Script Logging - “Execution Logs” subtab

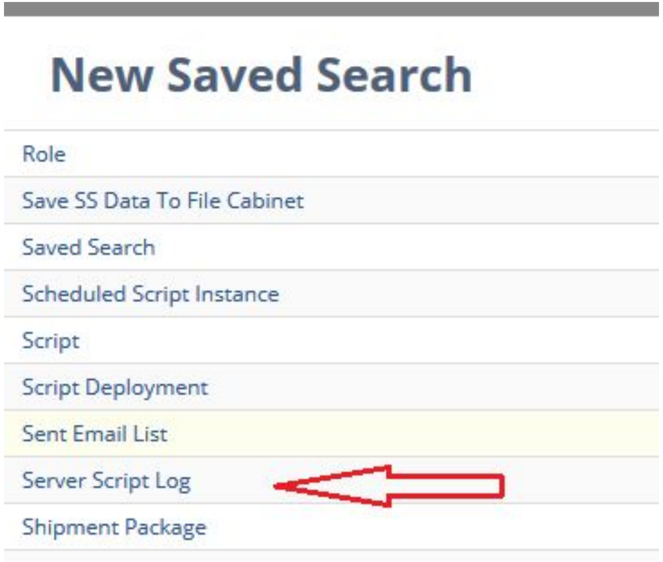
- Used often by developers
- See logs for all “deployments” of this script.
- Also visible from the “Script Deployment” record (only logs for that deployment)
- Only a 24 hour window of logs are displayed.



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.

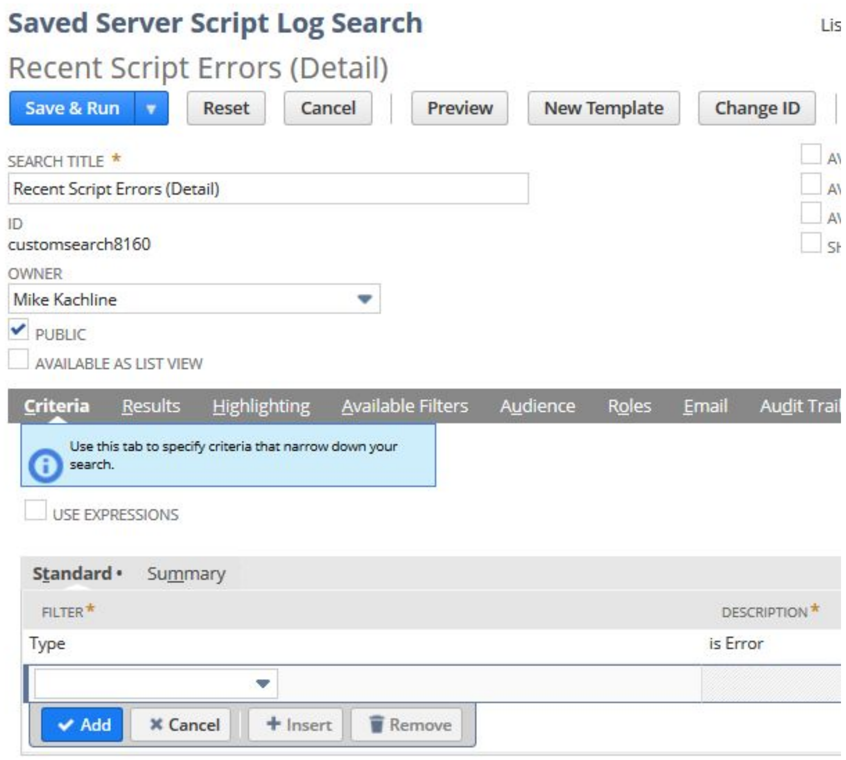
- Can be Emailed, setup as Portlet Dashboard
- Favorite Script Searches ...



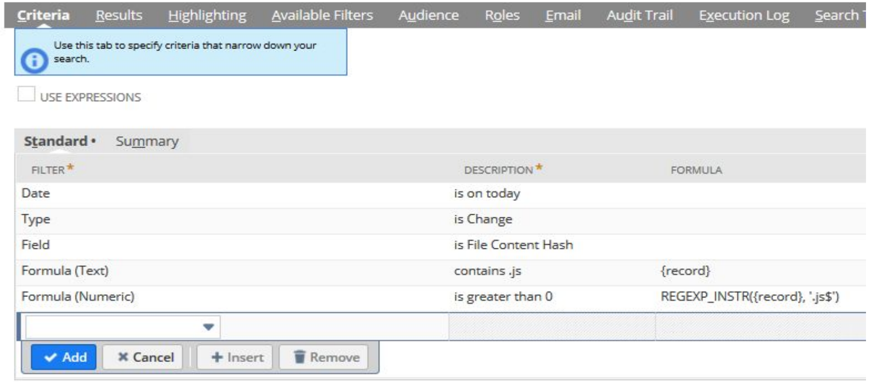
Favorite Logging Search: Recent Script Errors

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.



Favorite Search: Recently Changed Scripts



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.

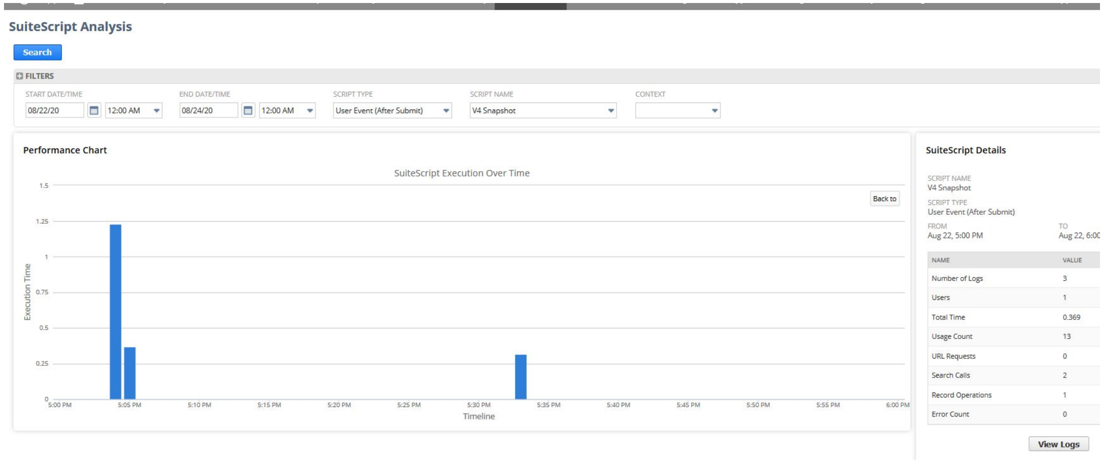
- "System Note" Search
- Criteria: "Field is 'File Content Hash'" / Date is today.
- Tells which script files have changed in the past day.

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

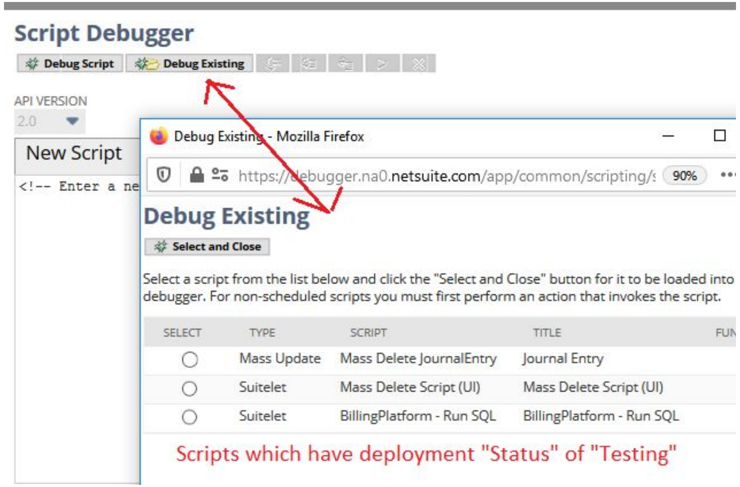


- Free Netsuite bundle ID=67350 “Application Performance Management”
- Identify which scripts are running slow, and why.

Noteworthy - Script Debugger

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

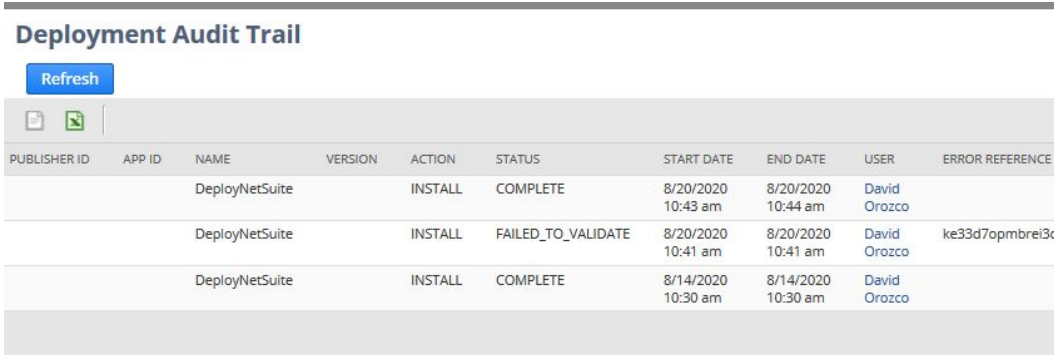
- Page: Script Debugger
- Offers standard debugger functionality (step in/over, viewing of variables real-time.)
- Does NOT work with Map/Reduce scripts.
- Can be used for “Ad-Hoc” scripts.
- “Existing” scripts must be in “Testing” status.



Noteworthy - SDF Projects

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

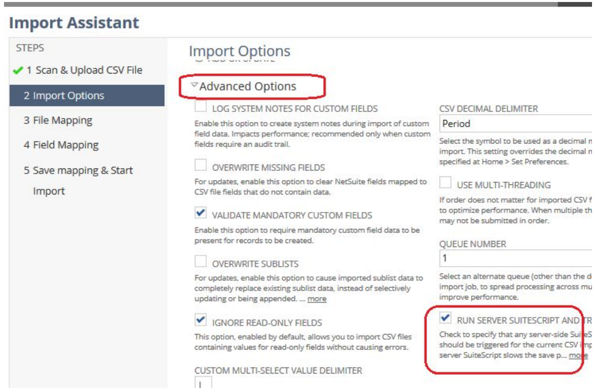
- Page: Deployment Audit Trail
- Think of these as “Desktop deployable bundles.”
- Often integrated with code “IDE” tool (Webstorm, Eclipse)
- Requires additional permissions



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	

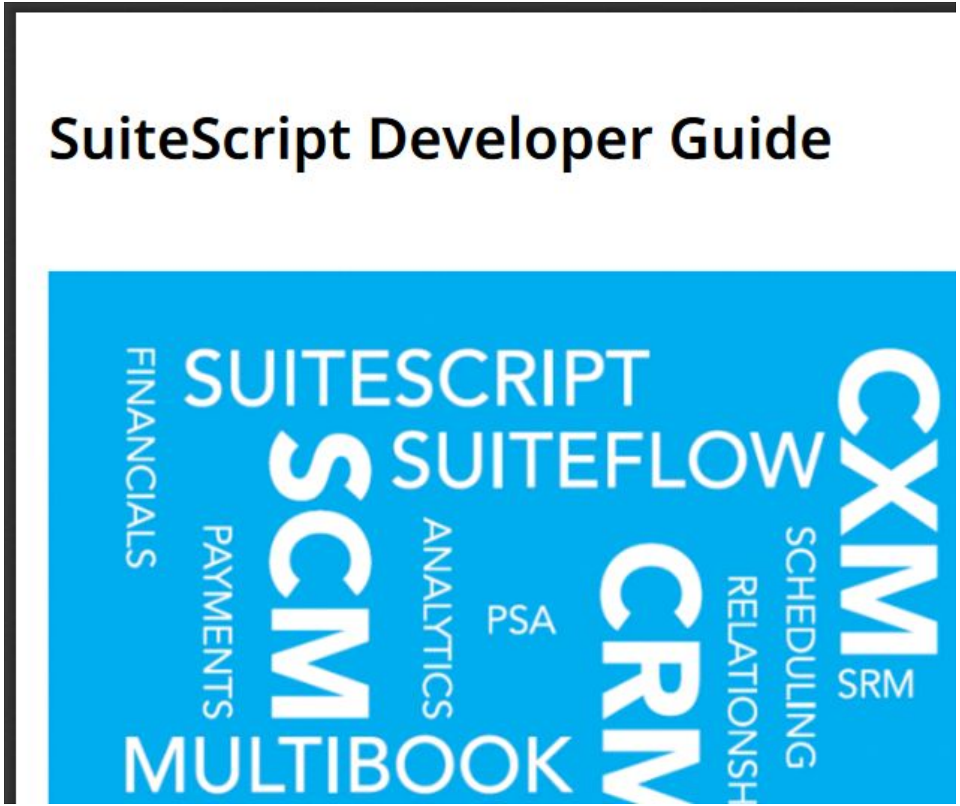
Script Caveats and Limitations

- “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”



Learning More

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



Thank You! - Questions?

RMNSUG - Scripting for Non-Scripters

CPE Credits?
Checkout Keyword: TREES