Welcome to Module Eight, Bulk Processing Part **One.** Now, in this module, we're going to look at how we schedule a script and how those scripts execute. We also want to know how we can create and deploy a scheduled script record. Now, unlike a client side script or a user event script where all you need to do is just deploy to a record and then have the user execute that, a scheduled script would need to be, well, scheduled, as the name implies, before you can have it run.

And then we're going to look at how we can execute the script immediately. So for example, if you have any ad hoc scripts, maybe you want to create something that you can run anytime you want that doesn't really have a schedule. For example, if you want to execute a script now, and then maybe two days later, and then maybe after a week, and then a day after that, then after three days. So if the schedule is just really erratic, and you want to run this manually, that's something that you can also do. So we're going to look at that.

So let's start the discussion by talking about scheduled scripts. So what are schedule scripts? So schedule scripts, again, as the name implies, is a script that you would trigger on schedule. Now, so far, the scripts that we've looked at would always get triggered by a user action. So if the user loads a record, that'll trigger your user event script, for example, your before load.

And then if the user edits or creates or copies your record, it'll also trigger your client side script, such as your page init. Or if the user does something to a form, either manipulating a field or manipulating a sub-list, that'll trigger your field change validate field, post sourcing, line init, and also the validate line. If the user saves the record. So again, this is always the user doing something to the record.

But what if we don't want any user interaction? We just want something that would trigger automatically. OK, so maybe I want to generate a report every start of the week. Or maybe you want to get a report at the end of the month. Or maybe you want to automatically send an email every end of the week or something like that or every other week or any schedule. So if you want your script to execute automatically, then you would use a scheduled script. So it's as simple as that.

Now, when you are going to create a scheduled script, there are a few considerations that you need to remember. Now, for one thing, using a scheduled script and setting that to a particular time, that doesn't mean that your script would execute at that exact time. So for example, if I create a script, and I schedule my script at 5:00 PM, it doesn't mean it's going to be executing exactly at 5:00 PM. What's going to happen is that it's actually going to be pushed to the queue at 5:00 PM, OK?

So what does that mean exactly? Now, every time you schedule a script, it's not going to put that on the actual execution based on the time. But instead, it's going to run the script one after the other based on a cue, where the system would always only run one script at the time.

So here, in this example, we can see that we have three scripts. We have a script A, script B, and also script C, which are all scheduled at 5:00. Now, here, even if the scripts are scheduled at 5:00, it doesn't mean that they would all run at 5:00.

And as you can see here, maybe script C is running at 5:00. Script B is running at 5:30. And then script A would be running at 5:09, OK? So that is possible. And again, that's because you can only run one script at the time, OK? That is at least for the regular account.

And the reason why we have the system is because we want to prevent users from accidentally scheduling several scripts at the same time. Because it is completely possible where a certain developer or maybe a group of developers would accidentally just create a schedule, not thinking about when other scripts would execute. Because normally, if people would be executing a script, they would be thinking, oh, maybe start of the day or end of the day, 9:00 or 5:00.

And just imagine if you have all users of all accounts. Remember, we are running the accounts on a shared server, meaning on one web application server, you have multiple accounts running on that application. So if you all run a script at the same time that will be using all the resources for the server. And that can potentially cause the server the crash.

Again, to prevent that from happening, we have a queuing system where all scripts would be pushed to the queue. On a regular account, you would have one queue for your execution. And you only have that one queue. So if you have several scripts, you have to wait for the first script to execute before your script would get executed. So you have to remember that behavior.

Now, if you have a larger company, and you think that you have several customs-- even for smaller companies that has heavy customization, and if you are going to use a lot of scheduled scripts, what you can do if you find this one queue limiting is that you can purchase a SuiteCloud Plus upgrade. With the SuiteCloud Plus upgrade, what happens is that you are given additional queues. So if you upgrade to level one of the SuiteCloud Plus upgrade, you are given five queues instead of one.

So that'll give you more than enough queues to execute your script. OK? So having five queues, that means that you can have like five scripts running at the same time. If you have level two upgrade, you can have 10 queues. And then if you have level three upgrade, you can have 15 queues.

Now, for most companies, level one upgrade would be more than enough. OK? Because even if you have dozens of scripts, having five queues available to you would allow you to basically just run at least five scripts at the same time, which I think, again, is just enough for most companies. OK?

So yeah, so just remember that if you are going to run a script, so if you have something on schedule, those scripts would not be triggered at the same time. And it'll always get triggered one after the other. If you want to have your script run all at the same time, you need to have multiple queues for that.

Now, the training account that we have right now, if you want to play around with that, that will be fine because we have level three upgrade for that account. So if you want to check, OK, what's going to happen if I move to this queue or to this queue or things like that, feel free to use the training account until it expires, OK? So just remember that the account will automatically expire after 30 days.

Another thing that you have to also consider when you are dealing with a scheduled script is that the schedule, like what I've said, does not necessarily mean that if you scheduled a script at, say, 5:00 it would be executed at 5:00. It just means that it would be open for execution at 5:00, OK? What happens with our system is that we have an application called a Scheduler. And that Scheduler runs every 30 minutes. OK? So that'll be 30 minutes after the last execution.

So for example, if you have a script set at 5:00, and then the Scheduler runs at 5:09, that means that your script would be pushed to the queue at 5:09. OK? So again, it's not going to be immediate. So those are the things that you have to consider regarding schedule.

One, it would be pushed to the queue at that particular time, but it doesn't mean that it would execute at that time. You have to wait for the other scripts that has been queued to finish before your script can execute. And two, you have to remember that the Scheduler only runs every 30 minutes.

So that means that your script, for example, if you set that to 5:00, it can potentially execute immediately at 5:00 if the Scheduler picks that up, or it can be executed at 5:30. So that's what happens with the Scheduler. So again, just make sure you are mindful of that behavior because with a scheduled script, people tend to think that whatever schedule they put in on the scheduled script, it would execute at that time, but that's not really the case.

Now, that we know how to create a saved search, we want to know how we can apply that to our script. Let's go ahead and build our scheduled script. Now, for this one, since we changed our DEFINE statement to a REQUIRE statement, let's just put that back. So I have my DEFINE statement here.

And for my function, instead of maybe creating a separate function for that, I'll just add that inside my RETURN statement because that'll be faster. So I have my execute function here. And function, I have my context object. And then put my implementation code. So everything here would be moved inside my execute function. That's it.

Let's see. Everything should still look fine here. OK, that's it. So that looks good. We don't have any missing braces or anything like that. Let me upload this.

And after uploading the script, I'll go to my NetSuite account. So I'll just go to Customization, Scripting, Scripts, New. And let's create our scheduled script. Let me look for my search here. So that's my search script. Let me create my script record.

Now, for this one, I'll just, again, just use the same naming convention, SuiteDreams SS, and then just call this escalated cases. Same internal ID \_sdr\_ss\_escalated\_cases. OK? Now, for the deployment, you have an option here if you go to your deployment record, you can create the deployment. Similar to any kind of script type, you have the title of your search, which I can use pretty much the same thing. I have my ID.

I also have some options here with the status. We're going to talk about that in a bit. The bug logs, you have your queue, and also all these other options. But instead of just using the Quick Deployment subtab, what I'm going to do is that I'm going to hover over Save, and then click on Save and Deploy so that we can go to the script deployment page and then look at the different options that we have. Because these options are-- they're OK, but they're fairly limited. Let's look at the full set of options that we have.

So anyway, so I have my case here. For my ID, again, just use the same ID. For my status, I have three different statuses here. So I have testing, not scheduled, and scheduled.

Now, testing is the same as any other script. It would only be executed by the owner of the account. You have scheduled. So if you set schedule, then whatever you put in on the schedule, that will be followed.

And then you also have not scheduled. So this is if you want to just park this scheduled script and not really execute it on schedule. This would be useful if you want to execute your script hoc.

For example, I'm going to run this now, and then maybe run this tomorrow and then the following week and then two days after that and then after a month. So if you don't have a specific schedule, then this would be perfect if you set that to not scheduled. Or if maybe it was previously scheduled, but you don't want to execute it anymore. So this could also be a way for you to disable your search.

Now, for us, I'm just going to set that to testing because I don't really need to execute my script on schedule, at least. Now, if I need to execute that on schedule, for example, for our use case, I'm going to set that to scheduled. And then, for this one, remember our scenario is that we want to execute that every Monday at 9:00 AM.

So I have different options here. So I have a single event if I want to just execute it when I want to execute that. If I want that to be a daily event, so I can execute that every weekday, or I can just have that execute every day or every other day, so if I change that number to two. I can also create a search where I want to execute that weekly, so every Friday. In our case, we want that to execute every Monday. And I want to execute that every Monday at 9:00, so 9:00 AM. So that's what I can do.

You can also execute that every other week. So I can change that, or every three weeks. So if you want to make that in a monthly event, I also have an option for that. Like, every 26th of the month or every fourth Friday of every two months. So those are options that I can change if I change that to two.

You can also have a yearly event. So as you can see, the settings are fairly straightforward. It's really easy to use. You have your start date, when it's going to execute first, start time. And you also have your end by.

So if you want this to execute forever, just put in no end date. But if you only would be running this until a specific period in time, then you can just open the calendar and just say, you know what, I can just execute that until May 24th, something like that. OK? So that's something that you can do.

In our case, we don't really need those information because we only have a single event execution. And then here, I'm just going to set this to testing. Then just execute it whenever I need. Or I can also set this to not schedule, OK? Because that's pretty safe because it's not going to execute on its own. It's just parked there.

Now, with our execution, you also have something here that should be interesting for you would be the queue. And that would be where you're going to push your scheduled script. So if you remember our discussion, you will need to set a queue on where your script would execute because it would not execute based on the schedule. It would be pushed to the queue based on your schedule.

So if you have other scripts that are scheduled before your script, then those scripts need to be executed first. Now, also, with your queue, if you only have one queue, for example, if you did not purchase a SuiteCloud Plus upgrade, then the option that you have here would just be one, OK? In fact, you wouldn't really have a dropdown. It would just have this grayed out.

Now, in our case, we have a level three SuiteCloud Plus upgrade, which means that it gives us 15 queues. So remember level one, five queues. Level two would be 10 queues. And the level three would be 15 queues.

If you want to assign this, you can assign that to any queue that you want. Do know that this does not automatically load balance. Because you might be thinking that, OK, you know what, I have 15 queues. I will just push that in queue one, and let the system manage that. That's not how it works.

If you have 15 queues, and all of your scripts are scheduled on queue one, then it's still going to be a bottleneck because all of your scripts are running on one queue. It is your responsibility to manage the different queues. What you can do is that you can just set like, for example, the first queue as your general queue.

Then maybe you could have like second and third queue for your reports. And then you can have like the fourth and the fifth queue for high priority scripts or something like that. It's just really up to you in how you want to manage that.

So talk to your development team if you have multiple queues. And say, OK, how are we going to balance our execution? So that's something that you need to think about.

Now, in our case, we don't really care because we have 15 queues, and we only have one script that we want to execute. So I'll just put that in queue one. Or I can put that in any queue that I want. So anyway, let me save this script deployment first.

And then once I save that, I'm going to edit it again. And by the way, I had to save it first so that I can put the changes on the database before I can execute my script. I'll hover over my Save button. Now, I have the option called Save and Execute. Again, if you don't see a Save and Execute as an option on your script deployment, save your script deployment record first. And then go back and edit it. I'll click on the Save and Execute button.

So after clicking on Save and Execute, it's going to push my script to the queue. And as you can see, it's already processing. And I'm on the top of the queue, so this would be my scheduled script status. If I have multiple scripts ahead of me, I will be seeing that on the list. So I can see here that I have like SuiteScript 2.0 scripts. It can also potentially have a combination of SuiteScript 2.0 and SuiteScript 1.0 scripts.

If I want to see all of the scripts that are scheduled, I can go to this page, which is going to be inside Customization, Scripting, and then you can go to Scheduled Script Status. So Scheduled Script Status would be the same page. But let me refresh this.

And then after refreshing, you can see here my status is complete. Looks like my script has completed processing. And by the way, when you see complete, that doesn't mean that your script successfully completed. It just means that it is done.

So it might have successfully executed, or it might have failed. So you have to make sure you check your execution. So here, if you have a long running script, it's going to tell you the percentage of the completion for your script.

Now, going back to our deployment record, remember we are just adding some logs. So if I go to my execution log, you can see here that looks like our script did execute properly because we were able to get all of our full results, similar to what we got from our previous module. So we have a shipping cost issue, an enhancement on the time input form, and then two installation issues. So that's it. So creating a script deployment for a scheduled script is very similar to other scripts, except that with a scheduled script, of course, you need to tell the system when you want to execute your script.

Some things to remember. Remember to schedule your scripts when the server load is lower. Our peak hours is considered at 6:00 AM to 6:00 PM Monday through Friday Pacific Standard Time. So make sure you schedule your script outside of that time so that it would not affect your execution.

Also, try to offload your processing to a scheduled script. And this is something that is used by a lot of developers. For example, if you have a user event script, and your user event script just processed a lot of information, what's going to happen is that, for example, if I save the record, I'm the user. Then after saving the record, it would process like information. As a user, I would have to wait for my user event script to finish before I can continue working on whatever I'm working on, OK?

So if the script would take like five minutes to execute, I'll be there in front of the computer waiting for that to finish. And that would be a very long time for a script to execute. And you don't want your users to wait more than five minutes. Or I think the average for a user interface design would be not have anybody wait for more than a minute. So people would just already start complaining if the script was running for more than a minute. So that's something that you want to avoid.

Now, to do that is that you can actually pass your processing. So instead of using a user event script, you can call a scheduled script. Then have the scheduled script process that. So what's going to happen is that when the user clicks on the Save button, it will transfer the processing to the scheduled script. The control goes back to the user. And then you can just have the user continue whatever they're doing while the script is running in the background.

So maybe if you want to send a notification to the user that the process has finished, you can send an email to say, OK, this is the summary of the things that were done. Or if it's generating a report of some sort, then it will just send an email. It'll tell you that, OK, this is the report that I've generated.

Let's see. Also, be mindful of the Scheduler's behavior. Remember that it would execute every 30 minutes, OK? If you schedule something, it doesn't really mean that it would execute immediately at that schedule. So it has to wait for the Scheduler to appear or to run before your script would get executed.

Now it's your turn. Please complete the following activity. This concludes Module Eight, Bulk Processing Part One.