**Welcome** to Module 11, Workflow Action Scripts. In this module, we're going to look at another way of automating NetSuite, and that would be by using workflows or by using SuiteFlow. We're also going to concentrate on how we can expand our workflow using workflow action scripts. And those are the things that we're going to talk about in this module.

So what we're going to cover in this module is that we're going to-- we'll at least understand what a workflow is all about. We are also going to start by configuring a simple workflow. And then we're going to extend that workflow by using a custom action script and also attach that custom action to the workflow.

Those who are new to workflows, workflow is a business process automation tool that allows you to create automations without having to write a single piece of code. Now, the thing with SuiteFlow is that you have to understand that it is built for being a business process automation tool, which means that you need to have a business process that you would be putting into the system for you to really use SuiteFlow in its full capabilities.

Now, one of the things that people think of when they think of workflow is that, you know what? I'm going to use workflow as an alternative to SuiteScript. But that's not really the case. So SuiteFlow was built, really, to allow administrators, or people, or power users, maybe, who know the business process but don't really know how to code and don't really know how to automate the business process. And again, that's really the main goal of SuiteFlow.

Now, in terms of SuiteFlow, again, there are a lot of things that you can do with SuiteFlow and SuiteScript that are the same. For example, there are certain automations that you can do with SuiteFlow that you can also do in SuiteScript. Now, you have to understand what the difference is between SuiteFlow and SuiteScript, and when you're going to use SuiteFlow and when you're going to use SuiteScript.

Looking at the main purpose of having SuiteFlow, that is a business process automation tool. Now, we want to use SuiteFlow if we have a business process that we want to put into the system so that the system would control that business process, OK? Compare that to SuiteScript. SuiteScript is meant to extend the capabilities of NetSuite, which means that if there are any functionalities that are not in NetSuite, then you can add those functionalities in so that you can have that on your account, OK?

Now, going back to SuiteFlow, one example of a use case for SuiteFlow is that, what if you have an approval process? So maybe you have an approval process in your company. And then what you do with the approval process, if you're not using an automation tool such as SuiteFlow, is that you will build a process document. Or maybe you will build a flowchart document. And then you're going to hand over those documents to your users or your employees and then have your employees follow that process.

Well, that is mostly OK. There are some instances in your process where your employees might not actually follow the process. So maybe there are some employees who keep forgetting to do certain steps of the process because maybe they're accustomed to doing something else before the process came out. Or maybe there might be, really, some delinquent employees who just refuse to follow the process because they want to do things their own way.

Now, if that's the case, then that's going to be a problem. Because if you look at it, the control is not really with the business process owner or whoever created the business process. But the control is with your employees. Because they can opt to follow or not follow that process.

Now, you can always have sanctions. But that's not really a good way of managing your business process. Now, looking at SuiteFlow as a business process automation tool, what you can do is that you can put your business process into the system and then have the system maintain or manage your business process. And this is going to be a win-win situation for you and also for the employee.

Because one, if you put the process into the system, then the employees would not have to think of the process. They wouldn't have to look at the document and say, OK, what do I need to do next? What are the things that I need to figure out? Do I need to create a form or whatever, fill something out?

Because if you put that in the system, the system would manage that for them. So if they need to submit something, you can just have something displayed on the screen and say, OK, I need to submit this document. Or if they need additional approval, then the system would automatically transfer the transaction to another person for approval. And then the employee does not need to do anything. That way, the employee can concentrate on their own function without having to worry about the process.

That's also good for you as the process owner if you built the workflow because the control is still with you. You're not passing the control to your users because you are directing the system-- OK, this is what's going to happen next. And this is what's going to be filled out. Or this is going to be a message that would appear on the screen. And again, you have full control on everything if you are dealing with a workflow. And again, that's what's SuiteFlow is about.

SuiteScript, on the other hand, since it's intended to extend your capabilities in NetSuite, it allows you to really build your own application. So if there's anything that's not there, you can build it. If you have enough resources to build that, then you can just go ahead and write the script, make your own project. You can integrate several applications together to make a whole, larger process.

With SuiteScript, though, like what we've seen, you need to make sure you understand JavaScript. Unlike with SuiteFlow, where if you are familiar with NetSuite, if you know the data structure of NetSuite, you should be able to build automation just by going to SuiteFlow. And really, they complement each other. It doesn't mean, that OK, I have to use SuiteFlow or I have to use SuiteScript. Because for developers, they kind of think that, OK, I know JavaScript anyway. So why would I want to use SuiteFlow?

But there are a lot of things that you can do with SuiteFlow that you can also do in SuiteScript. But it's so much easier to do that in SuiteFlow. Instead of programming for certain actions, you could just do a few clicks. And then your automation is done. Unlike with SuiteScript, you have the test. You have to make sure your script is working. You have to secure your script so that it doesn't collide with any other automation and things like that.

So again, as you can see here, these are very complementary even if there are some functionalities that are similar between the two. Now, before we start building our workflow, it is important for us to understand. What are the different elements of our workflow?

There are four basic elements of a workflow. We have a state, an action, a transition, and also a trigger. So as you can see here, this is an example workflow. And this is an approval workflow that approves furniture requests. So here, you have the Approver stage. And then it goes either Approved or Rejected. Or if a certain furniture has been recalled, they can also go to the Recall state.

OK. Now, let's try to look at what these elements are for. The first element is State. It defines where you're currently at in the business process. On its own, a state does not really do anything. So that is just a marker to tell you that, OK, I'm at this particular step. Because as we know, business processes are not just one step after the other. But you can jump from one step to the next step and then, maybe, jump two steps ahead depending on what happened to the process.

For example, you can have State 1. Or if you have a document, maybe you have a Section 1. And then you have a Section 5. And then you have Section 2 and then jump on to Section 10. Again, it really depends on your business process.

So a state will tell you where you're at. Are you on the first section? Are you on the tenth section, the fifth section, the sixth section, or whatnot? Now, aside from a state, you also have an action. And action is where the magic happens. So this is the actual automation in SuiteFlow.

An action would be whatever you want to do automatically or whatever you want the system to do automatically. For example, do you want to automatically send an email? Do you want to set a value? Do you want to transfer the user from one page to another? Do you want to initiate another workflow, or whatever you want to do?

And this is equivalent to statements. So whatever function you have in SuiteScript, that's something that would be similar with SuiteFlow. For example, if you use email.send to send an email, you have a Send Email action. Or if you want to use the Set Value function in SuiteScript, you also have a Set Value action in SuiteScript.

So Get Action would be the automation. The next one we have here is the transition. So transition would point you to where you want to go next. So for example, again, your business processing would jump from one state to another state or one section to another section. And a transition would tell you just where you are going to go next. So are you going to go through the first section to the tenth section, back to the fifth section, to the sixth section, then back to the third section, or whatnot?

Now, a transition is identified in the system or in SuiteFlow as an arrow. So as you can see, these arrows are our transitions. And then lastly, here, we have a trigger. And a trigger is simply a condition or an event that would trigger or that would execute a certain action or a certain transition.

So for example, you can see here, one of the conditions that we have is that if the status equals to Recalled, then that means that the control would come from State 1, and it would go down to State 4. You can think of triggers as your If and Else statement. So that if an Else statement would either execute an action, like maybe if a condition is met, it will send an email, or it would trigger a transition.

So if a condition is met-- let's say here, in this case, if the status equals to Recalled, then the transition would happen from State 1 to State 4. So those are the four different elements of SuiteFlow. So we have State, where you're currently at on your business process; Action, what you want to automate or what you want to do. Transition would be where you want to go next. And then Trigger would be the condition that has to be met so that an action or a transition would be executed.

In this walkthrough, what we're going to do is that we're going to create a simple workflow. Now, this would be a simple workflow. But the automation that we are going to use here can potentially be very useful for your user. Let's say you have a user that enters several expense reports a day, so maybe hundreds of expense reports a day.

And whenever you enter an expense report, what will happen is that once you save, you will be navigated to the same record. And there are a few keyboard clicks or mouse clicks that you have to press to go to a blank expense report. So what we want to do is we want to speed up the processing by automatically navigating the user to a blank expense report record. And then also, we want the default the same record that we've created from the previous expense report. So that's what we want to automate.

Again, just a very simple automation here, but this can potentially be saving a lot of time for users who are entering a lot of expense reports. But anyway, what we want to learn here is that we want to learn how we can enable the workflow feature. We want to create a workflow. And we also want to use the Go To Record action so that we can navigate the user from one page to another.

Now, at this point, we're not going to script just yet. What we're going to do is that we're going to create the base for our script by creating this simple workflow. Before we start creating our workflow, I just want to show you the default behavior if you're going to create an expense report.

So I'm going to go to Transactions, and then Employees, and then click on Enter Expense Reports to create an expense report. So for this one, let's say I want to create an expense report for, let's see, Kevin Abernathy. So I'll just use that.

And then here, and maybe I'll just have a hotel charge for, say, 300, maybe, US dollars, maybe just one expense there. I'll just click on Save. And then as soon as I click on Save, notice here that I am redirected to the same record for Kevin Abernathy.

And if I want to create a new one or another one, I can go to list, then wait for that to load a little bit. And then I can click on New Transaction and wait for that. And then if I want to create another one for Kevin Abernathy, I want to look for that record, and select that, and then go ahead and continue.

Now, as you can see, in between creation of expense reports, it takes me a few seconds to move from the previous expense report to the new expense report, let's just say about three to four seconds. Now, three to four seconds might not be a lot. But if you think about it, if I'm creating thousands upon thousands of expense reports a day, then three seconds saved per transaction, if you multiply that by 1,000, that's a lot of time saved. So by creating this automation, we really want to speed up the entry of our expense report. And we wanted to make it easier for our user to do that.

Now, before creating the expense report, also, we want to know how we can enable the setting. And that would be by going to Setup, Company, and then Enable Features-- again, that's Company, Setup, Enable Features. And then let me go to the SuiteCloud subtab.

Now, under the SuiteCloud subtab, this is where all of the SuiteCloud platform tools are available. For example, SuiteBuilder, you also have some options for SuiteScript. Now, for us, the option that we're looking for here would be SuiteFlow. So as you can see, SuiteFlow is already enabled.

And by the way, SuiteFlow, similar to SuiteScript, is default to every account. You don't need to pay for anything to get SuiteFlow. All they need to do is just enable that. Now, if you don't have that enabled, if you enable that, you will get a notification here. And all they need to do is just accept that.

If you scroll down, all the way down here, you have an option here that says, I Agree. So just click on I Agree. Make sure you read the terms of service. Because everybody reads the terms of service, of course. And you need to know what you're accepting. But anyway, this is pretty safe. But just click on I Agree. And click on Save. And that's that. That should have the workflow functionality.

So let's go ahead and build our workflow. So to build our workflow, you can go to Customization, Scripting, and then go to Workflows, New. Now, workflow is not exactly part of scripting. But this is part of automation. So that's why it's placed in this particular navigation path.

But anyway, let's build our workflow. So once we start building our workflow, you would notice here that this is very similar-- it's not exactly the same, but it's very similar to our script record. The first thing that you will be asked to do is give it the name. So let's say this would be my Expense Report Automation.

OK. For the ID, same ID name and convention, \_sdr\_ expense, or maybe exp\_rep\_automation to make it shorter. And then for the record types, as you can see, we're already deploying this to a record. So we want to deploy this to a transaction record. Specifically, we want to deploy this to our expense report, OK?

And then for your release status, you have the same release status here. So you have Not Running, Released, and Testing. So Released would be for everybody. Testing this would be for the owner of this workflow. And then Not Running would be a way for you to disable the script. If you don't want this workflow to be executed by your users, you can just set this to Not Running.

Now, for us, of course, we want this to be executed. So I'm going to change that setting. And if I scroll down here, I also have a few options. So off the bat, here, I have two options. And this option would tell me when I want to execute my workflow. Do I want to execute my workflow when I'm creating a new record? Or do I want to execute that when I am viewing or updating the record? Or in this case, do I want to execute that on both?

So for this one, I want that to be automated regardless if they're creating a new one or editing an existing one. So I'll put a checkmark on both. And then for the trigger type, if I open the Trigger Type, this should be very familiar to us. We have Before Record Load, Before Record Submit and After Record Submit. If you think about it, these are our user event script entry points.

When you load the record, you have Before Record Load. When you save a record, when it gets submitted to the database, we have Before Record Submit. And after saving to the database, you have your After Record Submit. Now, for this one, since we want this to be automating a value or navigating the user to another page, then I'm going to choose the After Record Submit event.

Now, for the event types, again, these are options that you can choose, for example, if you only want this to be triggered when they're creating a new record or, maybe, editing an existing record. So those things are things that you can set. But we're not going to set those values. So I'll just leave that blank so that this would be triggered regardless of what the user is doing as long as they're creating or editing a record.

Let me click on Save. And then after I click on Save, I will be given a workflow with one state. So if you're remember, a state would be an identifier on where you're currently at. Now, the first thing that I need to do here is that I need to rename my state so that it's descriptive of what I'm trying to do. Now, this state, since this is just a one-state workflow, I'm going to name this entry. So this would be my entry or my starter state.

This is where the execution would enter. And this would, maybe, branch out to other states on my workflow depending on how I expand this. Also notice what I did here. Instead of replacing the name, just remove that state one, what I did was that I added a colon. And then I added the word Entry. What is I do that?

Now, it's important for you to do that because you want to have a marker on where you're currently at on the workflow. If you have a longer workflow-- let's say a 20, or 30, or even a 50-state workflow-- it might be hard for you to troubleshoot if you don't know where you're at. Well, entry would be fine. Because if you look at the entry, you might be thinking, OK, that might be the first state.

But if you have like Validate Transaction, I mean, what state is that? If you have a 50-state workflow, how are you able to find Validate Transaction on that long list of workflows or states? It's going to be very hard to do that.

So what you want to do is that you want to have an identifier. You can have that State 1 as your identifier, or you can just have 1, or even S1. It's just really up to you as long as you have that identifier. So let me click on Save.

And then after clicking on Save, the next thing that I want to do is that I want to add my action. And action, again, is where the magic happens in a workflow because that's the automation that you're adding. So here, on the right pane, under the State subtab, I'm going to click on New Action. And by the way, you need to make sure that your state is selected. Because if you don't, then you don't have that State option.

So click on that state first, and then click on New Action. Then after that, look for the action that you want to do. And this is pretty much all of the actions that you can add. Now, compared to SuiteScript, where you have-- I mean, the list of modules alone is longer than the list of actions that you have in the workflow. So you can see there is so much more than you can do in a SuiteScript run compared to a workflow.

But here, now, what do we want to start? We just want to redirect the user from one page to another. So I'm going to choose Go To Record. And then on my Go To Record action, it's going to ask me, OK, where do I want to redirect to? For me, I'm going to redirect to an expense report. So I'll just have that expense report.

If I want to redirect to a particular expense report, I can choose that from my list. So these are fields inside my expense report record-- so for example, if I have an expense report, like Created From, if I had a basis for that expense report, like if I converted or transformed another record to an expense report or something like that. So I can just use that in the field option.

But for us, we want to redirect to a blank expense report. I'll just leave these options empty, OK? So let me click on Save. And that's it. OK. Now, if you think about it, if I tried to build this, it would take me lesson five minutes to finish everything. That's it. So that's how simple it would be to create a workflow. And this is a simple workflow.

But if you do something similar in SuiteScript, it might not take that fast. And it's not like SuiteScript, where you have to test, and you have to make sure everything is working, and you have to make sure that you've typed everything correctly and everything's properly deployed and configured. But here in the workflow, it's just a few keyboard, a few mouse clicks, and then you're done.

Now, let's go ahead and test this and see if this actually works. Let me go to a blank expense report. And then here, I will select an employee. Again, it doesn't really matter which employee I use. I'll try to use Rick Springfield. And then for the value, I'll just say, maybe, Dinner for $30. And then I have, maybe, a Breakfast for $20. That's fine. So we'll just have two entries.

Then as soon as I click on Save, remember the behavior on the default setting? It gave me the View mode of the record that I just saved. But now I'm being redirected to a blank expense report. I didn't even have to wait. As soon as the page loads, I am on my blank expense report.

Now, what we also want to do is that we want to automatically set a value. Where is Rick's record? There you go, Rick Springfield. Now, for this one-- and again, that's something that can potentially slow you down, especially if you have a very long list.

Now, we only have like a few employees on my account. But just imagine if you have, maybe, 5,000 employee records. Just finding one might take you long time. You'll keep scrolling down until you find what you need. So we also want to automate this so we don't load down with the selection of the employee record.

So I'll go back to my workflow and then open my Go To Record action. Now, here, at the bottom of my Go To Record action configuration, I have a sublist with several fields or with several columns. So for the first column, I have a Field column. And that column refers to any value that I want to set on my expense report. So this would be the fields on the expense report that I am redirecting to, my blank expense report.

So on my blank expense report, I want to set, say, my employee. And that would be set using my Entity Employee field. Now, I'm using Entity Employee field because the field that I have on my expense report-- now, if you remember, if I open that field-level help, this is actually the Entity field, OK? So that's why I'm choosing Entity.

Now, also, what I want to do is that I want to transfer the information from my old expense report. And I can do that by going to the Value field. So on the Value field, I'm also going to choose the same field. Now, you might be thinking, OK, why am I using two fields here? The reason why I'm using two fields here or two of the same field is that they're actually the same field, but they're referring to two different records.

The Value Field column is referring to the old record that you just saved, the previous one. And then I'm going to copy that value from the old record and transfer that value to the same field, but this time, to the new record that I'm redirecting to. And that's it.

So let me click on Add here. Let me click on Save. And let's see if this works. I'm just going to refresh my page to make sure that everything's properly set. And again, I'll just choose whatever. So I'll just choose James Rollings here. And then I'll have dinner for, maybe, $50.

And click on Add. And then click on Save. And again, we already know that our Go To Record works. But let's see here if we were able to get the value. And as you can see, we didn't have to look for James Rollings on our list, but that setting has already been preselected for us.

Again, this might be a simple automation, but this can potentially save a lot of time, especially for fields or records where you create multiple records per day. That's it for our walkthrough. On our next walkthrough, we're going to expand this and add a script to our workflow

What are custom actions? Our custom actions are scripts that you can attach to your workflow. Now, here, you can see that-- well, on the list of actions that you've seen when I was doing the demo, we can see that we only have a few actions available. And that means that there are a lot of things that you can't do with the workflow but you can do with a script.

Now, to give you a few of the common things that people want to do with a workflow but can't, one would be accessing a sublist. You can't currently access a sublist from a workflow, meaning you can't read a value of a workflow. You can't set a value of a workflow. And you're just really kind of stuck with you using your workflow.

So if you have any sublists that you want to access, you need to make sure you use a script. You can't call Web Services. So that's something that you also can't do. With a workflow, you can't update other records. You can update a record of the same type as your target record. But if you want to update another record, that's not something you can do.

For example, if I have an employee workflow, I can't update a task record. Or if I have a customer workflow, I can update a sales order record. I am limited to only updating the record of the same type as my workflow, OK? Same thing with initiating other workflows, you can only initiate the workflow if it is the same type. So if I have an employee workflow, I can initiate other employee workflows. But I can't initiate a task workflow.

Now, there are some strategies on how you can access, say, a sublist, or some strategies on how you can trigger a workflow. But again, you really need to understand the behavior for that to work. That's not really straightforward. It's more of a workaround than an actual possible solution. So if that's the case, the best thing that you can do is that you can just create a script.

Other complex business logic-- so if the business logic that you have is best served in a script, you might as well just do that in a script. Now, it doesn't mean that if you are going to build a workflow it's always going to be a simple workflow. Because I have seen really complex workflows that has really added very interesting solutions to the account because of that workflow. But again, compared to the amount of complexity that you can add or that you can handle using SuiteScript, I mean, this is really no contest. SuiteScript can handle a lot of processes, more than you can with workflow.

In this walkthrough, what we're going to do is that we're going to create a workflow action script so that we can extend the workflow that we built earlier. Here, what we're going to do is that first off, we want to pass a value from our workflow to our expense report. Now, this value could be any value. So you can see here that is a total value. So that's what we're going to use. But you're not limited to passing a number value. You can pass in any value that you want.

Also, we want to get the number of expense lines on our expense report. And as you know, you can't use or access a sublist from a workflow directly. So if you have anything that has an automation regarding a sublist, then you have to add a workflow action script to that.

We also want to update the expense report total that we've gotten from the workflow and also the number of expense lines, and then put that on our employee record. Again, as you know, you can't update another record that is outside of your target record on your workflow. For example, if you are creating an expense report, you can only update expense report records. Now, in this case, we're not updating an expense report record. But instead, we are updating an employee record. And again, it's something that you can do with a script but not with a workflow.

And then lastly here, we want to pass a value from our script to the workflow to inform the workflow if the action was successful or if it failed. Now, for the initial requirement, the total, that's a value that we're going to pass from the workflow to the script. But for this last requirement, we want to pass a value from the script to the workflow so that we can store that status value.

So what we also want to learn here is that we want to add in additional states so that we can see how we can control our transitions and also how we can process the result of an action. So what we want to learn here is that we want to learn how we can create a workflow action script and then how we can attach that to the workflow. So let's go ahead and build our script.

So let's build our script. Let me close this client-side script, and let's create our workflow action script. So for this one, I'm going to name it sdr\_wf for workflow, then update, and then employee. Now, for this one, we need two modules. I'm going to add my Record module so that I can load my employee record to perform an update. And I also need my Runtime module.

So as we've learned from the previous module, we need to use a Runtime module so that we can pass values from one script type to another. Or in this case, we're going to pass a value from our workflow to our script. So here, we have our own action function. So first thing that we're going to do is that we want to get the script parameter. We haven't deployed our record yet. So we don't have our script parameter. But let's go ahead and prepare for that.

So script, I'll just say workflowTotal. Similar to what we did in our previous module, we're not going to store the script format but get the value directly from our object. So here, runtime.getCurrentScript. And then we have our getParameter.

Then, for my getParameter method, I don't have the parameter name just yet because we haven't really created this. So I'll just leave the blank for now. And then next thing that I want to do is that I want to get a copy of my expense report record or whatever record the user is currently manipulating in this workflow. Similar to your user event script, what you can do to get your record reference is that you can get that from your Context object.

By the way, let me just rename this to Context. So I have my Context object. And then I can use the newRecord property. So the newRecord property would give me a reference to my expense report. And now that I have my expense report, I can do whatever I want with this record.

And these are things we've already learned. For example, if I want to get the expense lines' count, then I can just say-- maybe I'll just name this expenseCount. I can use my expense report record, then the getLineCount. So if you remember, we already used this before. And it'll just say sublistId. And I'm looking for my expense sublist, which is expense.

Again, nothing new-- and if we want to get my employee ID from my expense report, if you remember, we're getting that from my Entity field. So here, employeeId, and that'll be my expRep.getValue. Then I'm getting that from my Entity record.

Now, aside from that, we also want to get the-- or let's assemble the notes because we will be updating the employee record comments. So here, I'll just create Notes. And then I'll just say the notes would be Workflow Total. And then for my Workflow Total, I'm going to add the workflowTotal value, add in a line break.

And then I also want to get the-- also, we need to get here the number of expense lines. So I'll just name this Expense Count. There you go. Then on my Expense Count, I'll add in my expenseCount variable. Add yeah, that's pretty much it.

So what we're going to do is that we want to get the value from our workflow. So that'll be our Workflow Total. And then our Expense Count, which we are counting the number of expense lines. So we're not really totaling the expense lines. But we're just counting the number of expenses that's there.

Now after that, what we want to do is we want to update our employee record. Now, we already have the employeeId, or the internal ID of my employee record. So here, var employee. I can use the Record object, so Record module so that I can load the record. If you remember, the record.load function has two parameters-- the type, which determines what kind of record I want to load.

So here, I want to load an employee record. And I also want to load this specific employee record. That'll be my employeeId. And once I have loaded my employee record, the next thing that I'm going to do is just perform the update. That's it. So this would be just employee.setValue. And I want to set my comment section or my note section.

So if I go to an employee records-- so if I open this subtab, you can see here that my Notes script ID-- so I open that. It is Comments. So that's what we need to set. Here, I'll just say Comments. And then the Comments would be the Notes string that we have assembled here based on the value that we've gotten from the workflow and then the count of the expense reports that we've gotten from our expense report record.

Here, Notes-- and of course, I want to make sure that I save my changes, so employee.save. And that is it. We're done. So as you can see, there's really nothing new about our code. It's just that we're using a different script type for this particular code.

Everything here would be things that we've talked about. Like script parameters, we already talked about that. How you can access a new record, we talked about that early on. We talked about how you can count the number of expense lines.

We dealt with that with sublists. SetValue and getValue, we already know how to use that. record.load, so we talked about how we can use the Record module. And then record.save, for if you want to update a record or save a record that you've created.

Now, the thing that's new here is that I'm going to upload this. And this particular script would not be accessed via the record directly. But this would be attached to a workflow. So that's why it's called a workflow action script.

So to see how that works, let me go to my NetSuite account. And then on my NetSuite account-- and I'll just keep that open. And let me create a new script record. So let me look for my workflow action script here, click on Create Script Record.

Now, this is a little bit different configuration from what we're used to. So if you remember, we've been naming our scripts like this-- like SuiteDreams, and then the record type, and then a description of our script. But if we are going to use a workflow action, what we need to do is we need to use a more user-friendly name. Because whatever name we use on your workflow action, that will appear in the list of actions on your workflow. And you're going to see that later.

But the format that you want to follow here is that you want to use a verb, any description of what you're trying to do. For example, if you want to send an email, it'll be Send, then Email. Or if you want to validate certain values, you can just say Validate Transaction Total, or something like that, OK? So that's what we want to name our workflow action.

Now, in our case, I'm going to name this Update Employee Notes because that's what we are trying to do. So Update would be our verb, and then Employee Notes would be our description. For the naming convention for the ID, we're just going to use the same, pretty much sdr\_wf\_update\_emp\_notes.

Now, I'm going to go to my script parameter because I need to create a value for my workflow total. So here, I'll just create Workflow Total. Now, we're not going to total the value. And by the way, I'll change this to a Currency field.

Going back to what I was saying with the Workflow Total, we're actually not totalling any value here. But we are just looking at how we can pass a value from the workflow to our script. So that's the goal. We're just naming it Workflow Total. But this would be the repository value where the workflow would be passing the information to this value. And then, on our script, we're going to pick it up. And that's what we're doing.

So for the ID-- don't forget the ID. It's very important-- \_sdr\_workflow\_total. That's it. We'll click OK and go to my deployment record-- of course, deploy this to the same transaction. So I have to deploy that to my expense report.

If I deploy this to the wrong transaction record, then my workflow would not be able to recognize this workflow action. So this should be the same as your workflow deployment. So here, as you can see, I have deployed this to my expense report. And my script should also be deployed to an expense report.

So that's it. Let me click on Save. And I am done. Now, the thing that you have to remember here with workflow actions is that the person creating the workflow action might not be the same person who would be using that workflow action. Typically, what's going to happen is that if you have a separate set of workflow developers from SuiteScript developers, the workflow developer would be looking at the project and saying, you know what? There's a limitation on this project. I need to augment this a little bit. And I need to attach a script to it.

So the workflow developer would go to the SuiteScript developer and say, this is my requirement, please go ahead and build my custom action or workflow action. Now, after the SuiteScript developer builds out the workflow action, the SuiteScript developer would hand the control back to the workflow developer and say, OK, I've already built the workflow action. Go ahead and use that workflow action and test if this is working.

So that's what we're going to do. So here, I'm going to pretend that I'm the workflow developer. And as a workflow developer, I do not any programming. I know how to create the workflow, but I don't know how to code in Java or JavaScript, now, in this case, OK?

Now, here, what I'm going to do is that I will just go back to my workflow. And I'll click on New Action. And then when I go to New Action, if I scroll down, you would notice here that I have a new action that says Update Employee Notes. Remember, that's the name that we used on our script record. And it also has Custom to tell me that, OK, this is not part of the standard workflow package, but this was actually created by another user.

Now, for me, if I do not have any JavaScript programming experience, if I don't know SuiteScript, then I might be scared of using this-- like, OK, this is script, I don't know how to configure that. But if you open up Update Employee Notes action, as you can see, the configuration would be very similar to the configuration that you have on any standard action. So it's really no different.

Now, there are additional functionalities, of course, that this custom action provides. But if it's a custom action script, it doesn't mean that, for a workflow developer, you also need to learn how to script. That's not the case. And that's why it's very convenient for a workflow developer. Because once the SuiteScript developer hands the control over to your workflow developer, the configuration is very familiar.

Anyway, so let's go ahead and try to use our workflow action. So what I'm going to do here is I'm going to pass a value. And let's say for my workflow total, I'm going to use 500 because I want to pass the value 500 to my script. My script would be picking that up and then updating my employee notes.

Let's see if that's what happens. So let me click on Save here. There we go. I have my total. And then I'm going to go back to my script because I haven't gotten my script parameter yet. So I'll just copy that, paste that here to update my script, and then re-upload my script so that I can do the update.

OK. So let's go ahead and test this. Let me go to an expense report. Let me refresh this. And let's create a new expense report. Let's try Jim O'Brien's. And let's say hotel charge for, like, $500.

OK. Then let's add just one more, maybe breakfast for $20. Click on Add. And yeah, that looks good. So let me click on Save, and let's see if this works. OK. And it looks like our record was saved. And it was transferred to my blank expense report, meaning my Go To Record action worked.

But did my Update Employee Notes action work? Let's verify that by opening Jim O'Brien's record. So I'll just go through the global search and open Jim's record here. I'll open that in a new subtab. And as you can see here, on my Notes section for Jim O'Brien's record, we were able to properly get the value from our workflow, so 500.

And then for the expense count, it looks like I must have misspelled the sublist here. Because remember, -1, if you get that, that means that there's something wrong with the ID that you used. Now, in my case, I know that the Expense Reports sublist is scriptable. So it's probably just a misspelling. So let's go ahead and verify that.

Sublist-- yep, there you go. So expense-- I'm missing an N there. And that should fix it. But as you can see, we were able to properly access the information that we were expecting to access, like the value, from our workflow, and also the expense count. So maybe we'll just test that later as we expand the script.

So that's it. So that's how you would create a workflow action script, just create the script file. And you already know how to script for that. And then if you want to pass a value from the workflow to the script, you just want to create a script parameter.

Now, one more thing that we need to do here is that we want to pass a value from our script to the workflow. So how are we going to do that? So going back to my script here, what I'm going to do is that I want to confirm if the value or the record was updated or not. Now, the good thing about the Save method is that every time I call that, it would return the internal ID of the record that I have updated. If the update failed, then it will throw an error message.

So what I can do here is that I can just reuse my employee ID and then just say here, I have my employee ID, which equals to employee.save. And then I can add in a condition to say, if employeeId-- meaning, if it's not empty-- then what I want you to do is I want you to return Success or, maybe, put that inside single quotes.

Or maybe what I can do is that I can do the opposite. So if I can do not empty, so this would be a Failed. And then Success would be here at the bottom, return Success. Again, it's really up to you in how you want to do that as long as you have like a notification to say, OK, if this is successful, do this. If it's not successful, do something else.

So let's try this. Let me update this, upload this to the server. And one other thing that I have to change if I want to pass a value from my script to my workflow is that I need to change my script record and then inform my script record that I'm actually going to return a value.

Now, in this case, I will be returning a free-form text value. Now, under My Parameter subtab, I have a field here that says Return Type. That would tell me that this would actually be returning a value. So this, since I'm just returning a simple string, I'll just choose Free-Form Text.

So now my workflow can expect that this would be returning a value. Because right now, if I go to my Update Employee Notes, I can see here that, OK, it doesn't look like it would be returning a value. OK. But now, if I save this, we're informing the system that, OK, just expect a value from my script parameter.

Now, for my script parameter-- similar to what I've done when I pass the value from my workflow to my script, I didn't pass the value directly. What I did was that on my script, I created a script parameter and then used that script parameters to pass a value.

I'm going to do the same thing here. But instead of creating a script parameter, I'm going to create what we call a Workflow field. Now, a Workflow field is similar to a script parameter except a script parameter is attached to a script and a Workflow field is a field attached to a workflow. That's it.

So if I go to my workflow, and then on my workflow, I have a Field subtab or a Fields button, I can click on that Fields button and then click on New Workflow Field. So here, let me just create that Workflow field. The configuration, again, should be very familiar because this is exactly the same as your script parameter-- except, of course, this is for your workflow.

So here, I want to get the status. Remember, I am returning either Success or Failed. So for the status, I'll call this my Update Status. That should be a good enough name. So \_sdr\_update\_status. And for this one, this is just a free-form text, and I don't really need to configure anything else. Let me click on Save.

And now that I have a repository for the value that my script would be returning, I can go back to my state, then go to the workflow, then open my workflow action script here. And then on my Store Results In, I can say that, OK, I want to store that value inside my Update Status. Remember, before I change the setting-- remember, before, I set the return type here. When I went to my Store Results In, this was empty. There was nothing there. But now, since I'm expecting a value, I can store that value in any of these fields. Now, in this case, I just want to store that in the one that I've created, Update Status.

OK. Now, why is it important for us to have that update status? We're going to use that to extend our workflow so that we can know that if this workflow was successful, then we're going to transition on to a Success state. And if it failed, then it's going to transition onto the Failed state.

Now, if you were not familiar with the workflow, you might find this a little bit confusing. Because again, the system is a little bit different from scripting. But if you want to know more about the workflow,s I do recommend that you try to research that on SuiteAnswers or the Help Center. Or we do have a three-day Workflow for Developers course that you might be interested in if you really want to explore the capabilities of a workflow.

Because believe me, a workflow is just really powerful. It's not as powerful as SuiteScript, but it's still powerful nonetheless. And it's very easy to use. And that's really the key there. It's just easy to use. Whatever makes your life easier, right? So that's the thing that's important.

So anyway, let me build up my workflow by creating two more states. So here, I have State 2. And I'll click on the State button again here, then create State 3. Similar to what I did on my entry state, I'm going to rename this. So I have State 2. Maybe State 2 would be Success. And I'm going to rename State 3, and I will call this Failed.

There you go. And what I want to do here with my Success in a Failed state is that if the status that my script has returned-- if the status is Failed, meaning the update didn't happen, what I want to do is that on my workflow, I want to transition from State 1 to State 3. Or if my script returns Success, then I want to transition from State 1 to State 2.

Remember, the value that we are returning here is stored in a field. So on our Workflow field, we have our Update Status. So if I go back here, and then I can say here, whatever the result of my script, put that in your Update Status. So we know that the Update Status field would be populated with either Success or Failed.

Now, we can add in a condition on our workflow so that we can get the value of that update status. First, let's have a transition. So let's create the transition from State 1 to State 2 and State 1 to State 3. So here, if I want to create a transition, all I need to do is just hover over the state where I want to transition from.

So for example, if I hover over State, I should see a transition handle. But as you can see here, it doesn't look like the transition handle is working. And this can potentially happen to you. If you hover over a state and you don't see the transition handles, that means that it's just that the workspace did not load properly. You don't need to change any settings, or you don't need to enable or disable anything to get the transition handles.

So go ahead and refresh the page. If I go back here, as you can see, I get the transition handles. And by the way, the transition handle is simply just a half-circle. That's it. So grab that half-circle, and then drop that to the state that you want to transition to.

So I'll just create the transition from State 1 to State 2 and then just create another one from State 1 to State 3. And that's it. So it's either going to go left, Success, or right, Failed. Now, that would be controlled, again, by my Update Status field. So what I can do is that I can double-click on this transition. So here, double-click on that.

And then on the right side, I have a Condition section. So if I open my Condition, I can say that-- so if you think about it, this is kind of similar to your search where you have your field. You have your compare type or operator. And then you have your value.

So what I'm going to do is just say, OK, if my Update Status equals to Success, then go ahead and perform the transition. So let me save that. And then we'll pretty much do the same thing on the other side. I just want to make sure I use the right setting, so Success and Failed. OK.

So I double-click on my transition here. And then I do pretty much the same thing. But this, time my Update Status, if it equals to Failed, then go ahead and move to State 3. Click on Save. And that looks good.

So let's see if this works. Let me go to an expense report here. Let me refresh my expense report lists. Let me close those. We don't need that anymore. And let's see. Let's try to use June's record. And I'll just create one. So dinner for, say, $40. Again, it doesn't really matter what value we use here.

But I'll just go ahead and save, and let's see if this works. OK. There you go. So you can see everything is still working. If I verify June's record, if I open June's record, I can see here on my note-- so yeah, it looks like it is working now. So instead of giving me -1, it's giving me 1. So I have gotten the correct value. And then we also passed the same value from our workflow, which is 500.

Now, that's not exactly what we were looking for. What we are looking for is that, OK, did we transition properly from State 1 entry to Success or Failed? Now, with a workflow, unlike a script where you have just a general script that is running on the background that you can use to track all of the executions regardless of the record, a workflow always be attached to one single record.

So if you want to know what happened to that workflow, you have to open the record that executed the workflow. This is not the record that executed the workflow. This is the result that we redirected to. So I'm going to go to my Recent Records and then open the last one that I've created-- so this one, Expense Report by June.

And if I open that, what I can do is that I can go to System Information. Then under System Information, you can see these are the things that were changed on the account. I can look at my active workflows if I have several workflows that are currently running for this record. But what we're interested in would be our workflow history.

So if I look at Workflow History, I can see what happened to any workflow, running or not, that executed for this record. And I can see here that, OK, it started in State 1. And then it was successful and then moved on to State 2.

I can even verify the logs. So in the link here, if I click on the log, you can see here that, OK, I entered that the workflow. So this state initialised or initiated my workflow. And I have two actions. I have custom action, which we've created. And then we have a Go To Record action, which we've added to the workflow.

If I open my custom action, you can see here that, OK, it looks like the workflow is passing the value 500 to my script. And my script returned the result of Success. Now, since this was successful, I can also open the transition. So you can see here, the transition has a condition, if the update status is equal to Success. And we can see that it was successful because the result of that condition is True. Then that's why it transitioned to our State 2, Success.

So again, that's just a very high-level overview of a workflow. There are a lot of things that needs to be discussed with the workflow. But we're not really concentrating on SuiteFlow right now. Because what we need to learn is about workflow action scripts and how we can attach a workflow action script to a workflow. But again, if you're very interested in a workflow, I do highly recommend that you research that or attend our SuiteFlow for Developers course.

So that's it for this walkthrough. That is about it. Just a few things here, make sure that you are going to create a custom action only when necessary. Don't go overboard and create custom actions for everything. If you can do your automation using a standard action, stick with a standard action. Because the more scripts that you create, the more things that you would have to maintain as a SuiteScript developer. So make your lives easier by making sure to tell your workflow developers to say, OK, I think you can do this with a standard action. Just stick with a standard action.

Also, when naming your custom actions, like what we did, use a user-friendly name. So instead of using SuiteDreams, wf, something, something, then what you can do is that you can just use the format verb and then description. So let's say if you want to create a custom action for sending a campaign, maybe, your script would be Send Store Campaign or something like that, OK?

Also, make sure the scope of your project before you're starting. And again, this is something that I've seen a lot happen in the wild where non-developers are really excited to build their workflow because they don't need to program anything. But if you really don't follow the best practices, it would be hard for you to maintain the workflow. So just make sure you understand the workflow before jumping in.

Understand the scope of your project so that you know which tool you're going to use. If the majority of your automation should be happening on the script, then why are you going to force the system to use a workflow and then augment that with a workflow action script, right? So you just might as well just start with the script.

And then if there's anything else that you want to quickly perform, then you can add a workflow, maybe workflow one or two workflows here and there. But if the majority of your project can be handled by a workflow-- say, 80% to 90%-- then stick with the workflow. And then for the small things that you can't handle with the workflow, go ahead and add a workflow action script.

And then lastly here, we do have a course that you can take if you are a developer. Please consider looking at the SuiteFlow for Developers course, which allows you to understand how workflow really works. Because there are a lot of configurations that you need to consider-- for example, the different execution cycle, things like that. But there are a lot of things that you can learn from SuiteFlow. And knowing two different tools for automation, like SuiteFlow and SuiteScript, would really help you in building your project. So instead of just sticking to one automation tool, having two tools would be even better.

Now it's your turn. Please complete the following activity. This concludes Module 11, Workflow Action Scripts.