Call scripting is a highly valuable and immediately rewarding way to use Clicktools. It allows for rapid ROI and can vastly improve user experience in Salesforce™ by saving time and improving data quality.

We will look at two examples, one simple and one slightly more complex to illustrate the time-savings possible with call scripting in Clicktools.

The basic activity flow goes like this:
- A customer calls in or an agent calls out.
- The script changes based on the conversation or experience of the agent.
- Clicktools updates the relevant information.
- The agent moves on to the next (inbound or outbound) call.

A simple inbound script

The first example we will look at is based on an inbound call. The script may be delivered by clicking on a custom link against the contact or in larger organizations, could be popped by your CATI system.

An example script is shown on the next page. See this script without the pre-populated information by clicking here.
Example in-bound call script when first deployed from within Salesforce.

First, you will notice that the call script is skinned to look like any other Salesforce information.

Second, you will see that we have passed information into the script from Salesforce. In this example, we have passed across the Contact’s name as well as her email address and phone number.

In this example, we’re collecting fairly standard information contained in an Opportunity, but obviously, in your implementation, you would create questions to support your particular sales process and product(s).

The completed example in-bound call script with updated email and phone plus opportunity details.
When the information is collected on the call and the Submit button is pressed Clicktools automatically:

1. Creates the Opportunity

![The created opportunity.](image1)

2. Records the completed call and relates it to the Opportunity and Contact

![The completed activity recorded against the Contact and Opportunity.](image2)

3. Adds the Contact as the Decision Maker to the Opportunity

![The Contact Role recorded.](image3)

4. Updates the Contact record

![The Contact record with the updated telephone number.](image4)
Again, with Clicktools, all of this happens automatically upon Submit. How long do you think manually completing these tasks might take? Let’s look at the information we have touched in Salesforce:

Clearly, Clicktools drastically reduces your administration overhead when used for even the simplest call script, but what about a more complex script?

In the first example, we did something quite straightforward, creating a few things and linking them all together, which although simple, clearly saves a lot of time and effort.

But what happens if we want to implement an outbound script based around a telesales campaign? This is far more complex but, nevertheless, a very common use case.

This example call script is used for cold-calling to a new lead or to promote an offer or product to existing Leads and/or Customers.

So, let’s look at the example script. Again, we launch the script from within Salesforce, but this time, as we are collecting more information, the script is more dynamic, meaning that the questions presented will change based upon the answers given. Note that you could also split the script across multiple pages, if you prefer.

Using Clicktools for simple call scripting, processing 10 orders a day (at 6 minutes each) saves 366 hours a year – that’s around a month and a half’s work a year!

1 PERSON COMPLETING 10 SCRIPTS A DAY TAKING 6 MINUTES TO UPDATE INFORMATION IN CRM SAVES:

NOTE: You don’t have to initiate the script from within your CRM. If you use outside agencies or call centers to make sales or marketing calls you can give them the script directly – they can go through the scripts and the information will be added immediately to CRM. Hurrah! No more spreadsheet manipulation and waiting on telesales results.
Depending on the individual’s answers to this script, the actions we need to record in Salesforce are different. Let’s look at some alternative actions we may need to record and, as before, think about the manual effort required to record these actions in Salesforce.

<table>
<thead>
<tr>
<th>Result of Call</th>
<th>Needed in CRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you did not get through to the contact:</td>
<td>Record Activity as Failed Call</td>
</tr>
<tr>
<td></td>
<td>Update Campaign Status to reflect “Failed – call back”</td>
</tr>
<tr>
<td>Contact expresses interest now:</td>
<td>Create a High Priority Task for the Account Owner</td>
</tr>
<tr>
<td></td>
<td>Update Campaign Status to reflect “Immediate follow-up”</td>
</tr>
<tr>
<td></td>
<td>Create Opportunity for Account Owner</td>
</tr>
<tr>
<td></td>
<td>Add Feed to Chatter explaining HOT Lead</td>
</tr>
<tr>
<td>Contact expresses future interest:</td>
<td>Update current Campaign Status to reflect relevant “Follow-up campaign”</td>
</tr>
<tr>
<td></td>
<td>Add Feed to Chatter explaining HOT Lead</td>
</tr>
<tr>
<td>Contact is not interested and doesn’t want any more calls:</td>
<td>Update the relevant Opt-out field (standard or custom) to stop future calls</td>
</tr>
<tr>
<td></td>
<td>Update the Campaign Status to reflect “Failed – no more calls”</td>
</tr>
<tr>
<td>Contact is not interested but happy to receive future calls:</td>
<td>Update the Campaign Status to reflect “No Interest”</td>
</tr>
<tr>
<td></td>
<td>Update Contact to reflect No Call status</td>
</tr>
</tbody>
</table>

Now, you not only have to record the individual actions, you have the additional manual effort of ensuring the responses are recorded in the correct way and added to the right campaign. Sure, it’s not a complex task, but it’s mind-numbingly dull, takes a lot of time, and is prone to error (and we are talking about salespeople doing this)!
Again, Clicktools can do all of this automatically. Let’s look at a sample completed script followed by the way Clicktools records the response in Salesforce.

Let’s assume the script starts positively and we are put through. When the agent selects the “Put through” option, the script expands to show the next question.

The same thing happens as we progress through the call, so the Agent only sees questions that need to be answered and are relevant to the call. This is amazingly powerful and allows you to change the script based on products sold, features requested, or even the experience of the Agent!
Let’s assume Daisy is interested. In this case, the Agent is prompted to close the follow-up call.

The script ensures that the right details are captured and provides an area for Daisy to record any notes.

As soon as the Submit button is pressed, Clicktools goes into action. As you can see, this was a HOT response so, according to our rules above, Clicktools does the following:

- Updates the campaign to reflect the need for immediate follow-up
- Records the call in the Activity History and allocates the Account Manager a high priority follow-up task
Adds the Opportunity – note how the Campaign Influence is recorded to ensure we can track back the win (or loss) of this job to the telesales team.

We set a reminder for the Account Manager task. As soon as the task is recorded, a pop-up appears telling him to call the Contact. So, whether the Agent makes the call from a faraway land or from an outsourced telesales agency, the Account Manager knows about the Opportunity immediately.

The pop-up to annoy the account owner.
And, finally, we can put a post into the Chatter feed, alerting everyone who follows Daisy.

This example may seem complex, but it is pretty commonplace for lots of Salesforce customers. Using Clicktools drastically reduces the amount of time to update information and, more importantly, enables you to respond faster to opportunities. Plus, it protects against collecting incomplete or incorrect information, if left in the hands of salespeople, who may not be accustomed to performing such detailed work. Your telesales agency may do a great job of gathering information, but they may not be allowed direct access to your CRM system, so you’ll need to deal with manually manipulating and importing data periodically after the telesales campaign, which is certainly not ideal in terms of timely response.

IF YOU ELIMINATED THESE MANUAL UPDATES, EACH AGENT WOULD SAVE:

3.3 HOURS A DAY    16.6 HOURS A WEEK    3 DAYS A MONTH    > 1 MONTH A YEAR

Using Clicktools can make your agents virtually TWICE as productive. The time spent making updates to CRM could be used making calls!
We have shown a simple and complex example of a sales-based call script, but there are many other examples where the concept of scripting can pay dividends.

Why don’t you try one today?

Some examples include:

- Traditional phone-based Market Research (internal and external)
- Sales qualification calls
- Sales follow-up calls
- Inbound Support calls
- Outbound Support calls
- Customer onboarding set-up calls

Now, It’s Your Turn.

We hope this information gives you good insight into how you’d benefit from implementing a call scripting process using Clicktools and CRM. We hope you get started as soon as possible. When in doubt, remember this core principle; when you strategically collect, centralize, and act on customer interactions, you will dramatically improve customer relationships. It’s just a matter of integrating Clicktools with your existing CRM solution, which once you get a hang of it, is one of the most powerful ways to positively affect your business. And the best part – we are here to help every step of the way!