Software by Inisoft

syntelate XA

Getting Started Tutorial

Published 27 March 2024

Release 2.12.0

Contents

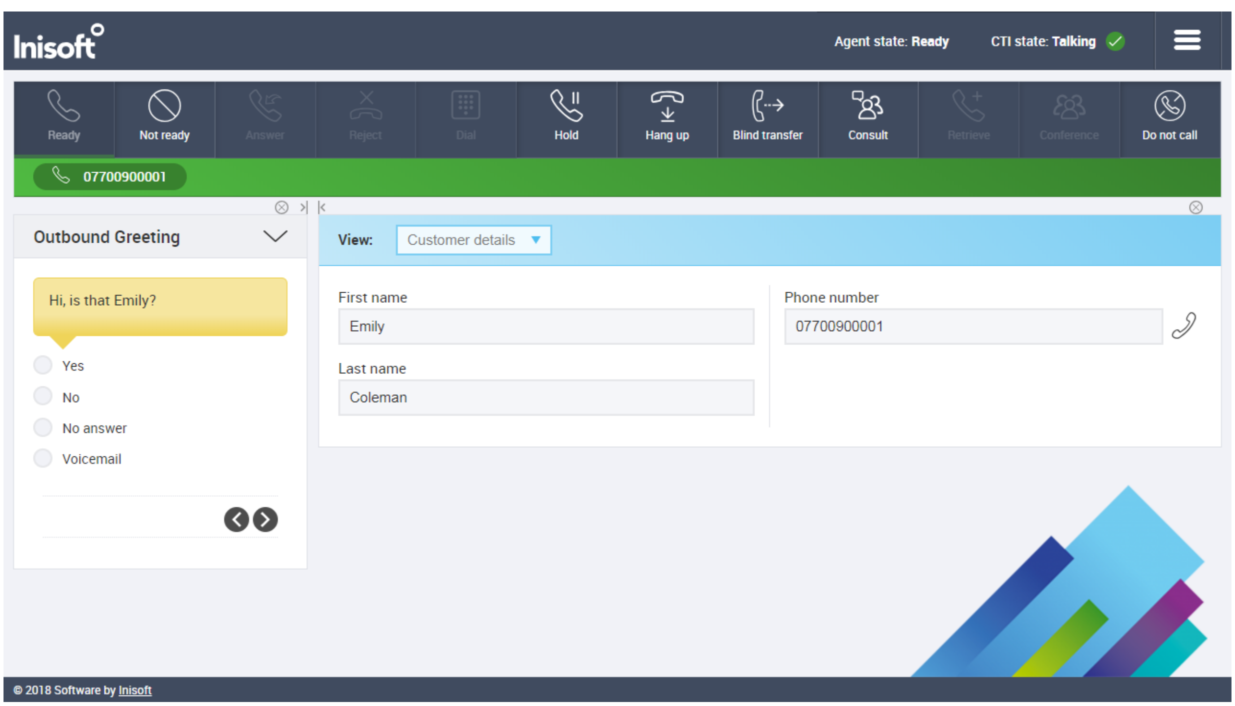
1. Desktop Design Tutorial

This tutorial is designed for people completely new to Syntelate XA and who are looking to learn how to design and use agent desktops.

This tutorial was designed to be used with a Syntelate XA environment that has been specially set up to support this tutorial. Among other things, this environment should include a universe called **Agent Desktop** and also be set up with dummy outbound records.

Note: If you do not have access to such an environment, you can still read through this tutorial to learn the basics of Syntelate XA. Be aware, however, that you may not be able to follow some steps exactly.

By going through this tutorial, you'll pick up the basics of Syntelate XA and create a very simple agent desktop that looks like this:



Before you go any further, you'll need a few things:

- The URL for a Designer environment that you can use with this tutorial
- The URL for the corresponding runtime environment
- Login details

You may not know what a "Designer environment" or a "runtime environment" is yet, but that's OK: we'll cover that in the tutorial!

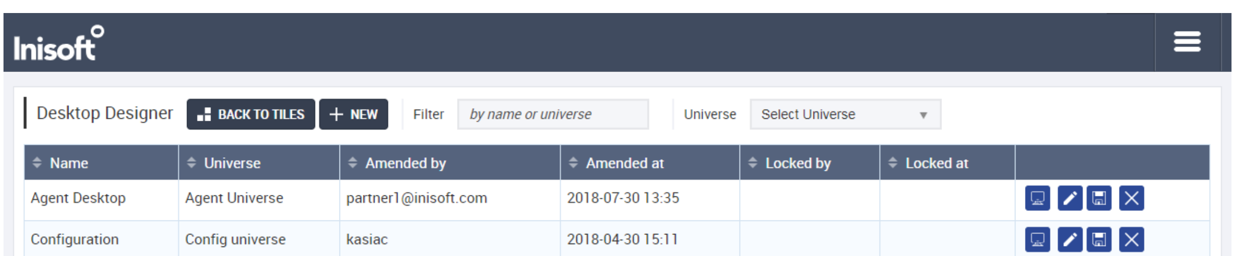
Without any further ado, let's get started by creating a desktop.

2. Creating a Desktop

In your web browser, visit the URL for the Designer. This is where you can design desktops for your contact center agents.

Log in using your login details. You'll see some tiles:

You're going to create a new desktop, so click **Desktop Designer**. You'll be shown a list of existing desktops:

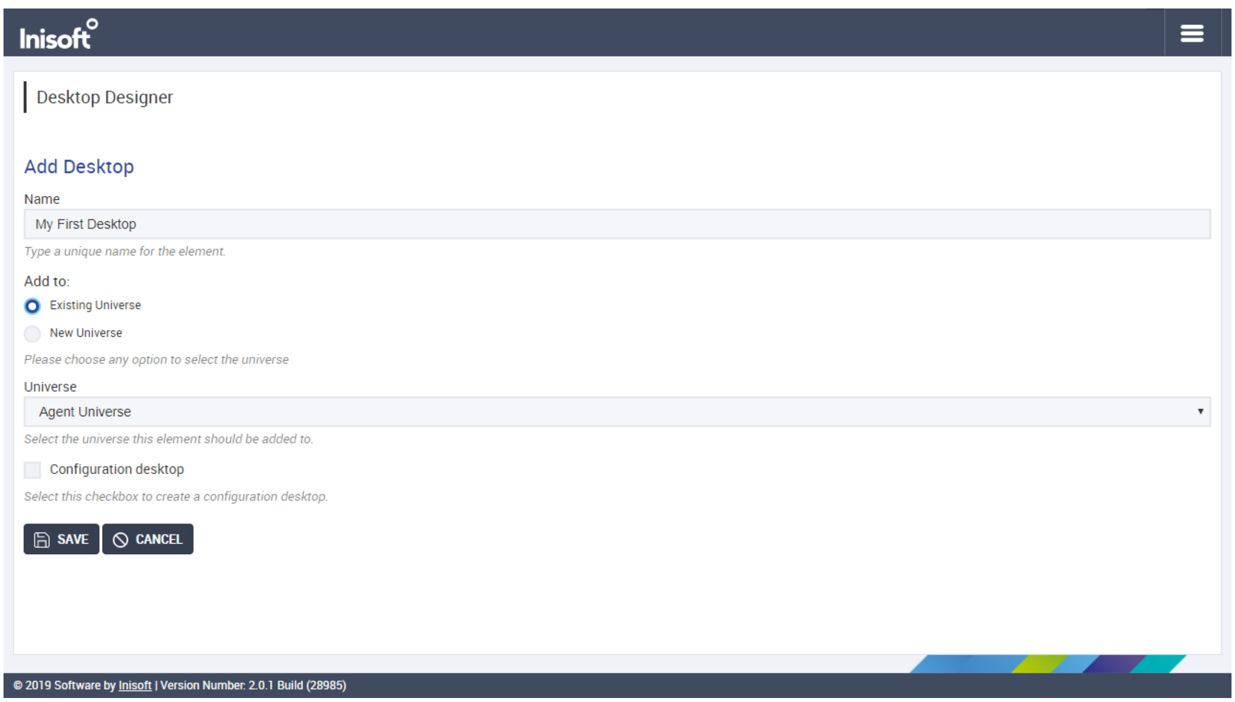


Click **+ NEW**.

You'll be asked to type a name for the desktop. Give it a name like **My First Desktop**.

Note: If that name is already in use, call it something similar instead. This applies throughout this tutorial, whenever you're asked to name something.

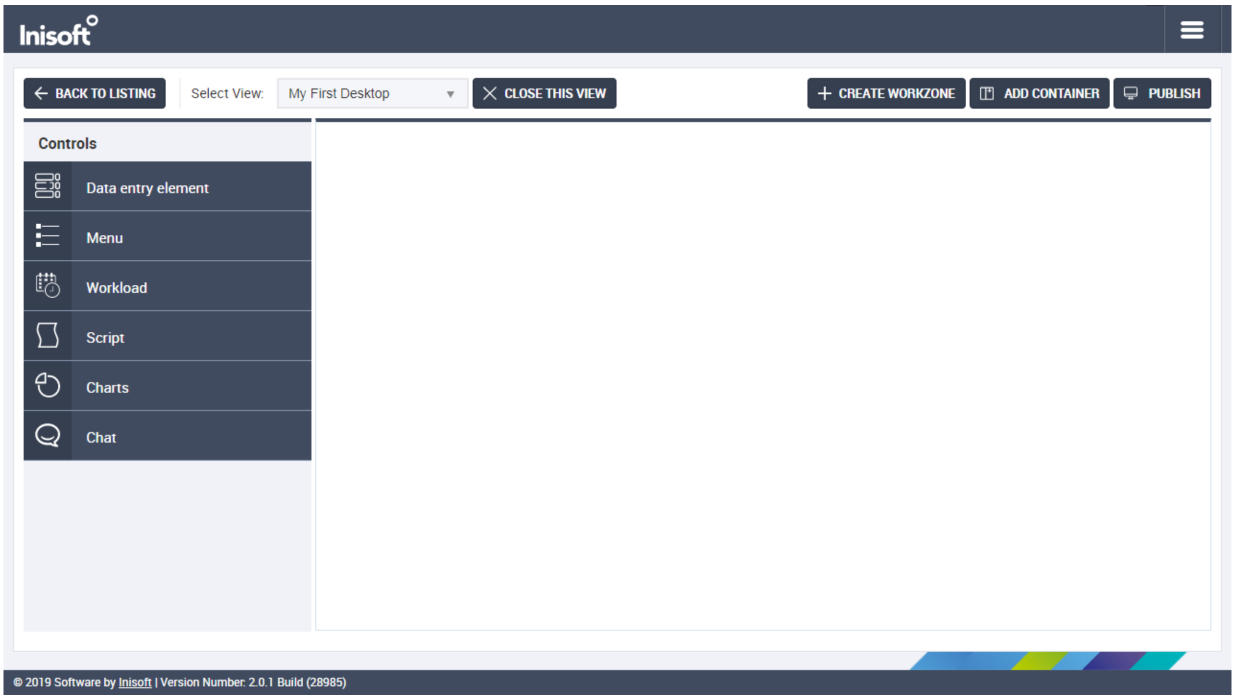
Next, choose to add the desktop to an existing universe and select **Agent Universe** as the universe. You're probably wondering what a universe it, but we'll talk about that later. For now, let's push on!



Note: If you're following this tutorial on an environment that has *not* been specially set up to support your learning, the **Agent Universe** universe may not be available. If this is the case, you'll have to use a different universe.

That's all you need to enter on this page, so click SAVE to create your desktop.

You'll see a blank desktop:



In the next chapter, we'll start designing the desktop, but first click CREATE WORKZONE.

As with universes, we'll talk about workzones later, so don't worry about this for now.

Now let's move onto designing a basic desktop!

3. Designing a Desktop

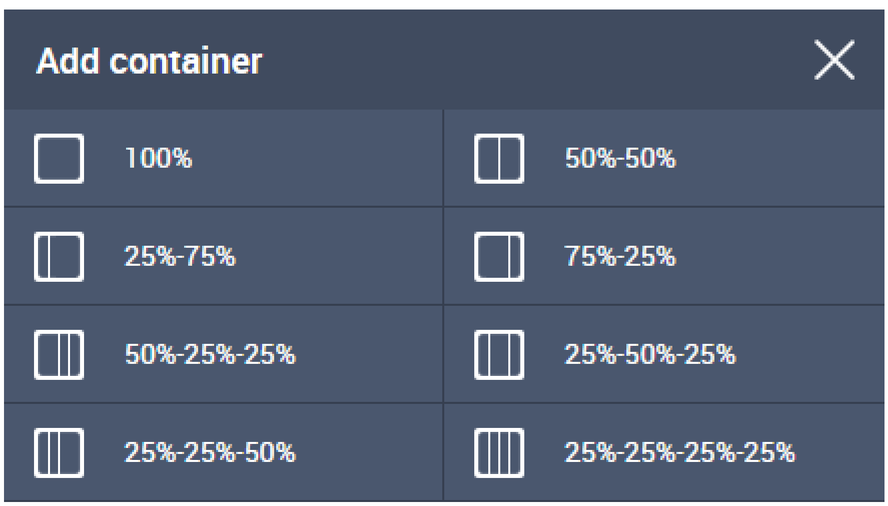
It's time to design our desktop. The **desktop**, of course, is the interface that agents see when handling calls or other types of interaction.

3.1 Adding a Container

The first thing you need to do is add a **container** to your desktop. A container lets you organize desktop controls into columns. This will make sense once you see it, so let's just go ahead and do it.

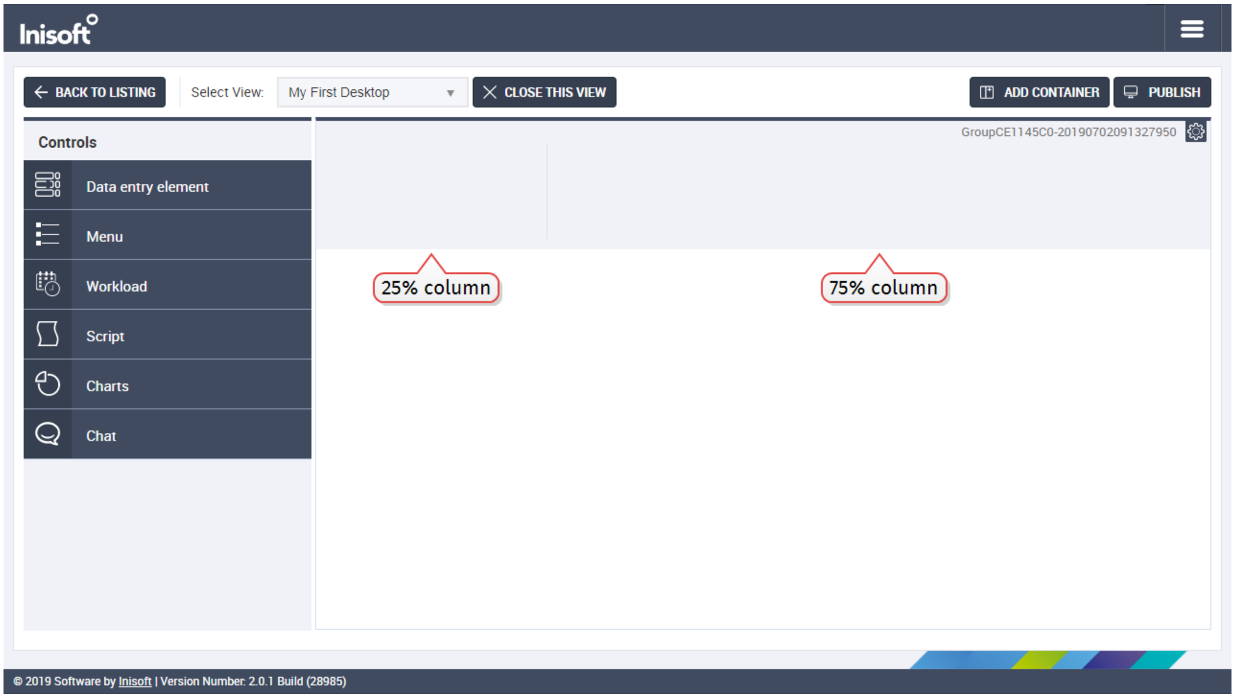
Click 

. You'll be presented with various choices:



Click **25%-75%**.

A two-column container is added to your desktop. The left-hand column of this container will take up a quarter of the horizontal space on the agent's desktop, while the right-hand column will take up three-quarters of the space.

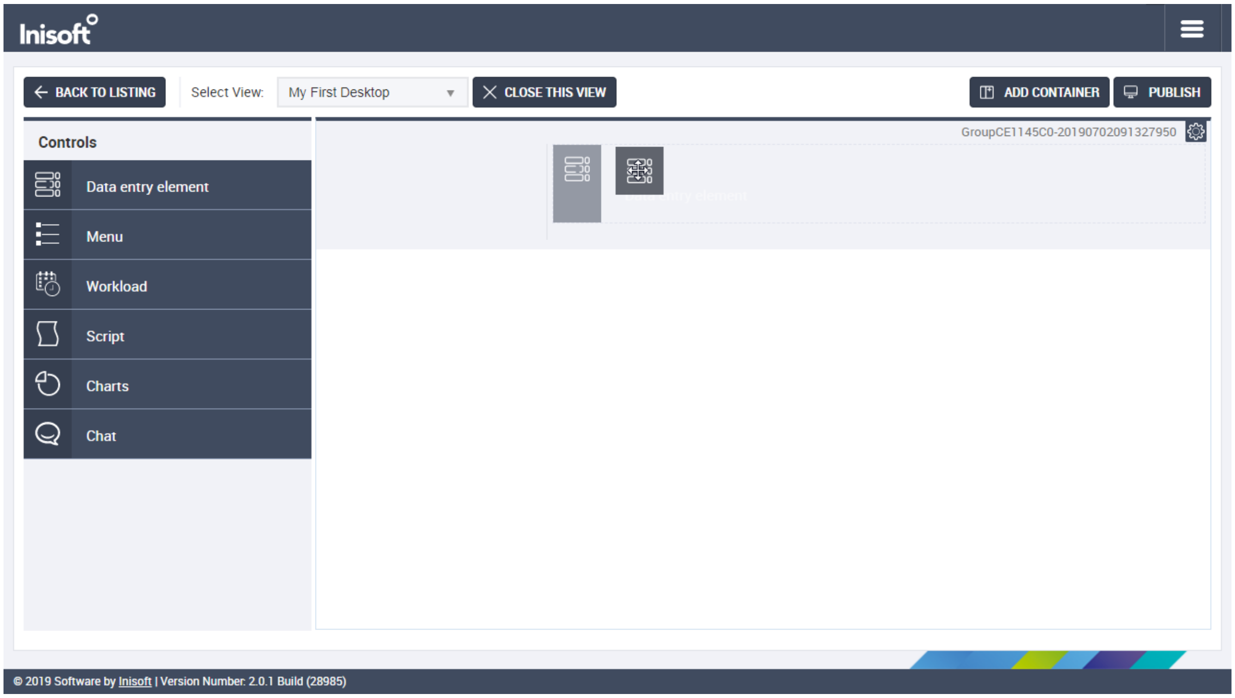


3.2 Adding a Data Entry Element

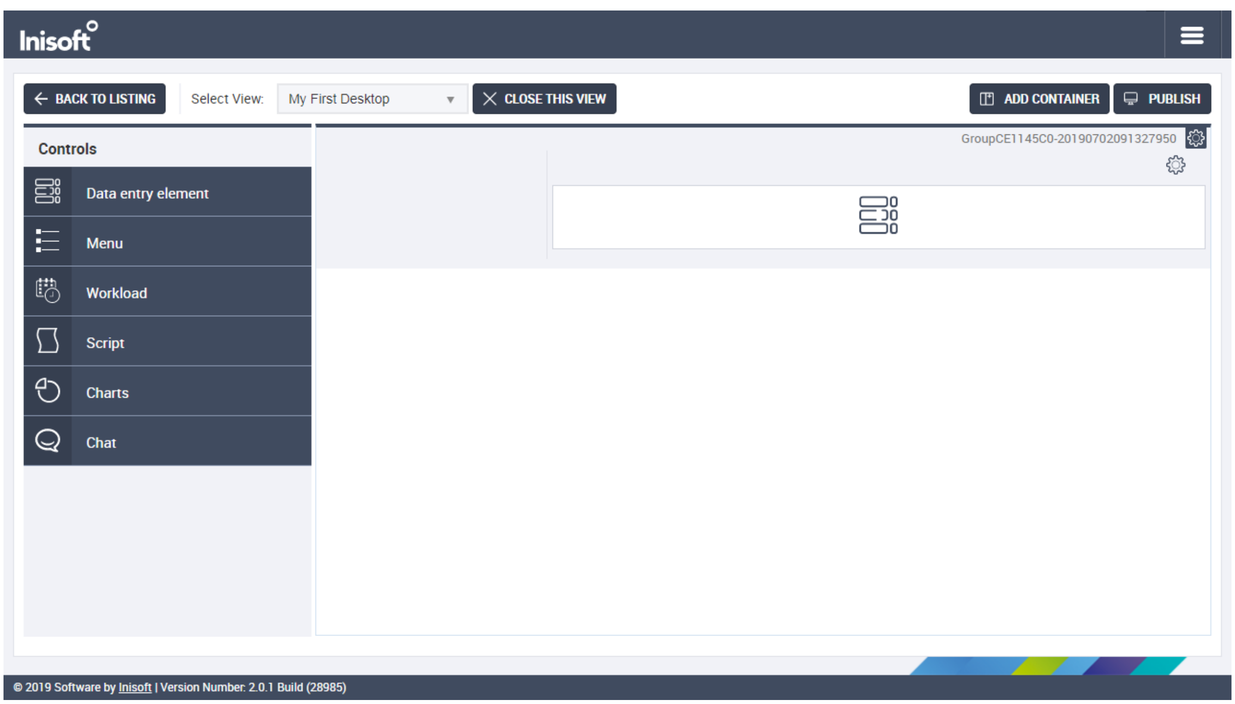
Now let's add a control to our desktop: we'll add a data entry element.

A **data entry element** lets agents enter and view details, such as the customer's information. It actually also has other uses, but let's not get ahead of ourselves – we can talk about those later.

In the left-hand **Controls** menu, click and drag a data entry element to the right-hand column of your container.



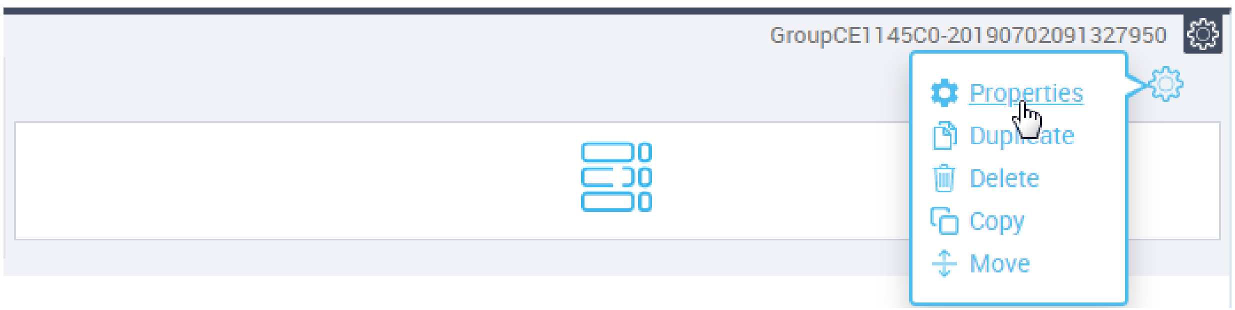
When you let go, an empty data entry element is added to your desktop.



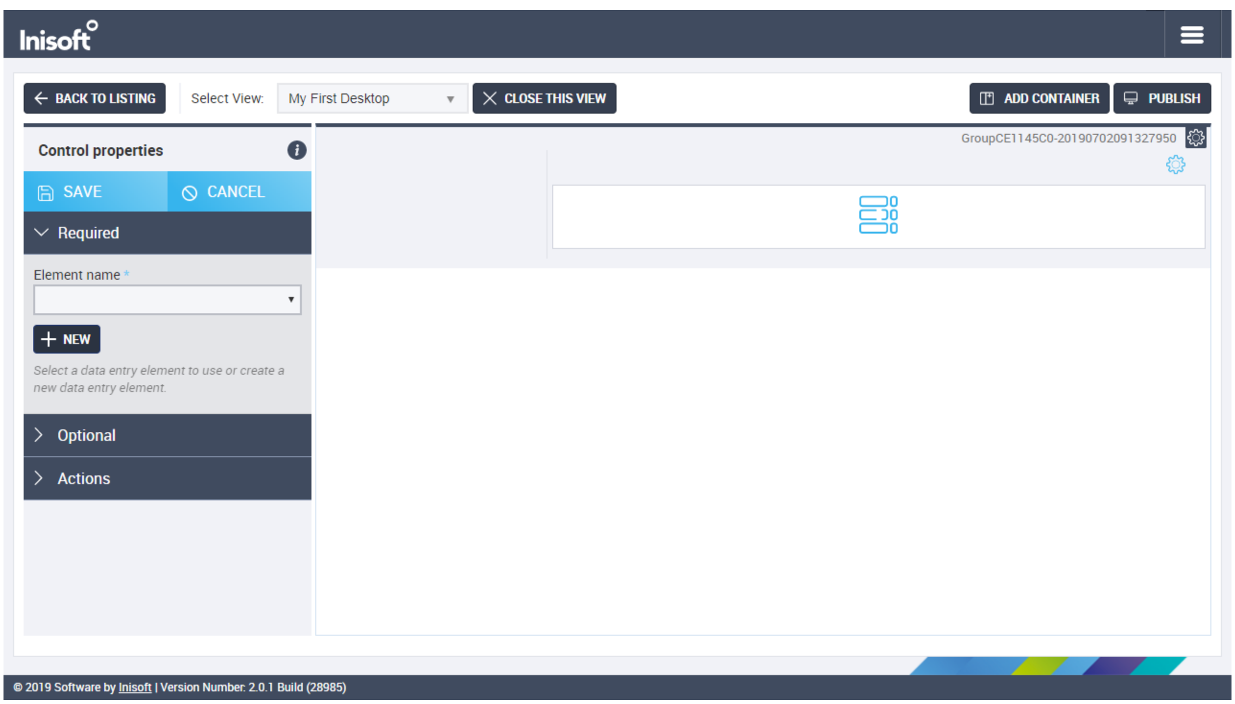
3.3 Viewing a Data Entry Element

We now have to set the data entry element that we want to use. We do this by editing the control's properties.

Click the cogwheel on the data entry element to show a menu, and then click **Properties**:



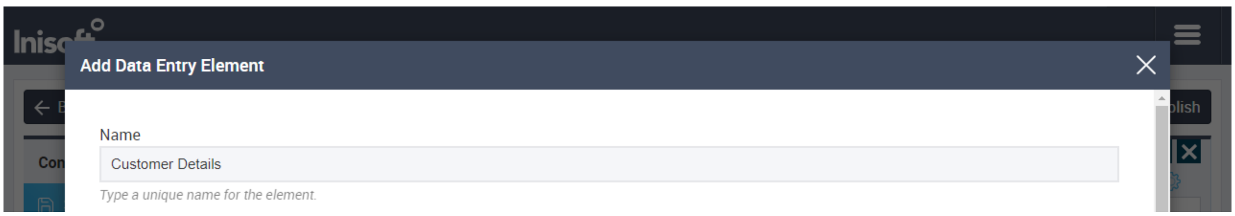
The left-hand **Controls** menu is replaced by a new **Control properties** menu:



In the **Element name** dropdown, you can select an existing data entry element, but let's not do that – let's create a new one!

Click .

You'll be asked to type a name for the data entry element. We're going to use this data entry element to hold customer details, so type **Customer Details** (or something similar).



Next click **SAVE**. You'll be asked whether you want to view the data entry element now. Click **YES**.

A new tab will open in your browser, and an empty data entry element will show.

You've now jumped from the **Desktop Designer** (which you selected at the very beginning) to the **Data Entry Designer**. You'll notice that the interface looks very similar – that's to keep things simple!

In the next chapter, we'll look at designing a data entry element.

Note: Some browsers block pop-ups, which can prevent your browser from opening the Data Entry Designer in a new tab. If you're having problems, check your browser's settings.

4. Designing a Data Entry Element

You're looking at an empty data entry element. Let's add some stuff to it!

4.1 Adding a Container

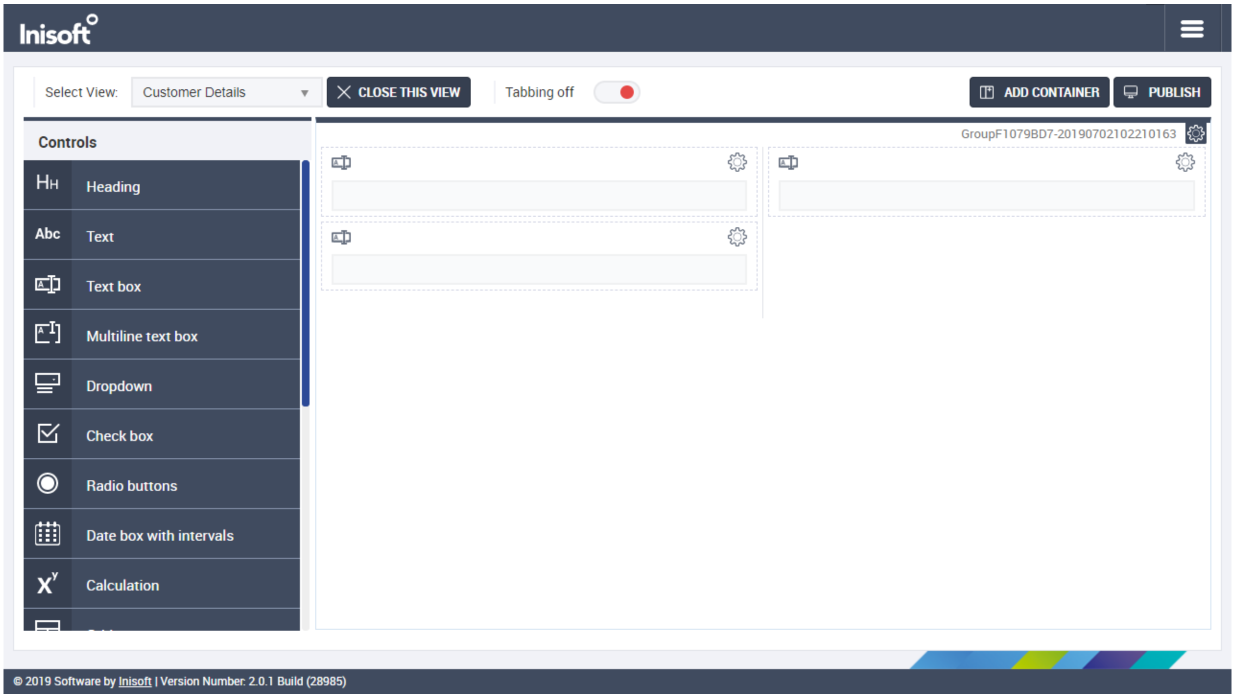
Just as with a desktop, the first thing we need to add to a data entry element is a container.

Remember, just click 

at the top of the screen. This time, select a **50%-50%** container.

4.2 Adding Text Boxes

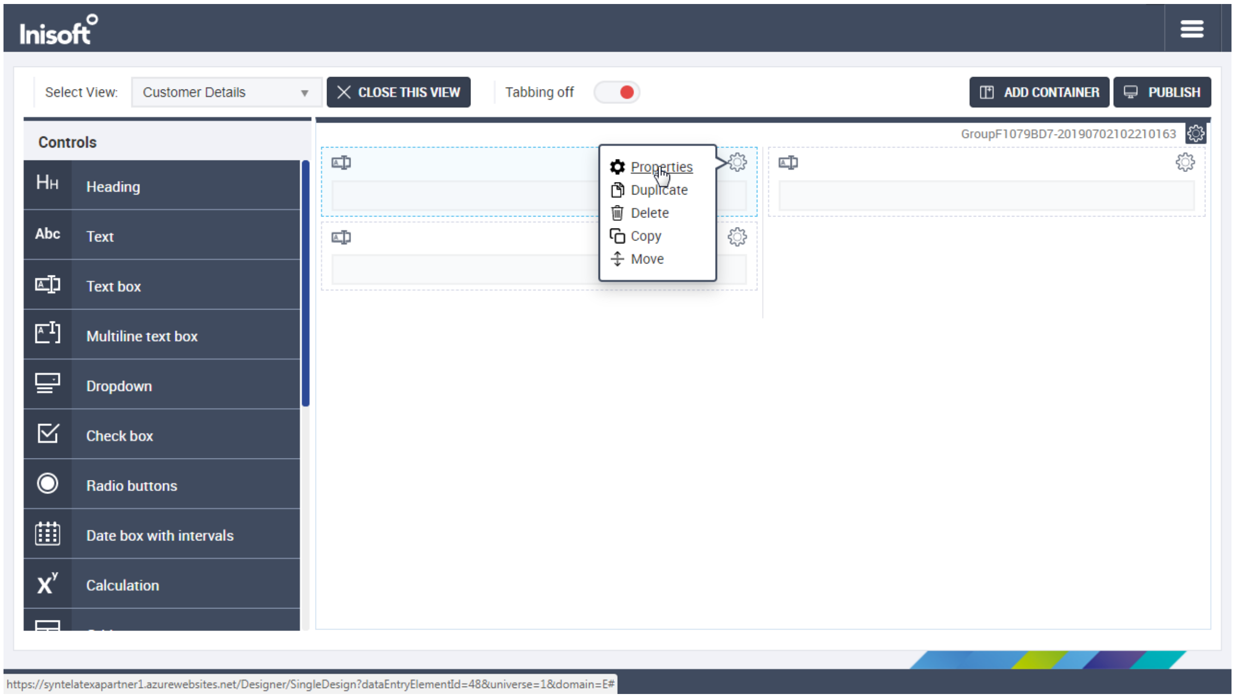
Now drag three text box controls (*not* text controls!) to your container so that it looks like this:



We're going to use these text boxes to hold:

- The customer's first name
- The customer's surname
- The customer's phone number

Let's go! Click the cogwheel of the first text box and select **Properties**.



Just like in the Desktop Designer, the left-hand **Controls** menu is replaced by a **Control properties** menu.

The first two properties for a text box let you link the text box to a database field. All right, we've been putting it off, but now it's time to say a little about universes.

4.3 About Universes

Syntelate XA lets you manage one or more **data universes**.

Think about it this way. Every company that uses Syntelate XA will need to store different kinds of data. For example, a retail company stores very different information about their customers than a bank.

A universe defines things such as the database tables that the agent desktop can access. If you use Syntelate XA with a separate customer relationship management (CRM) system, this is defined in the universe too.

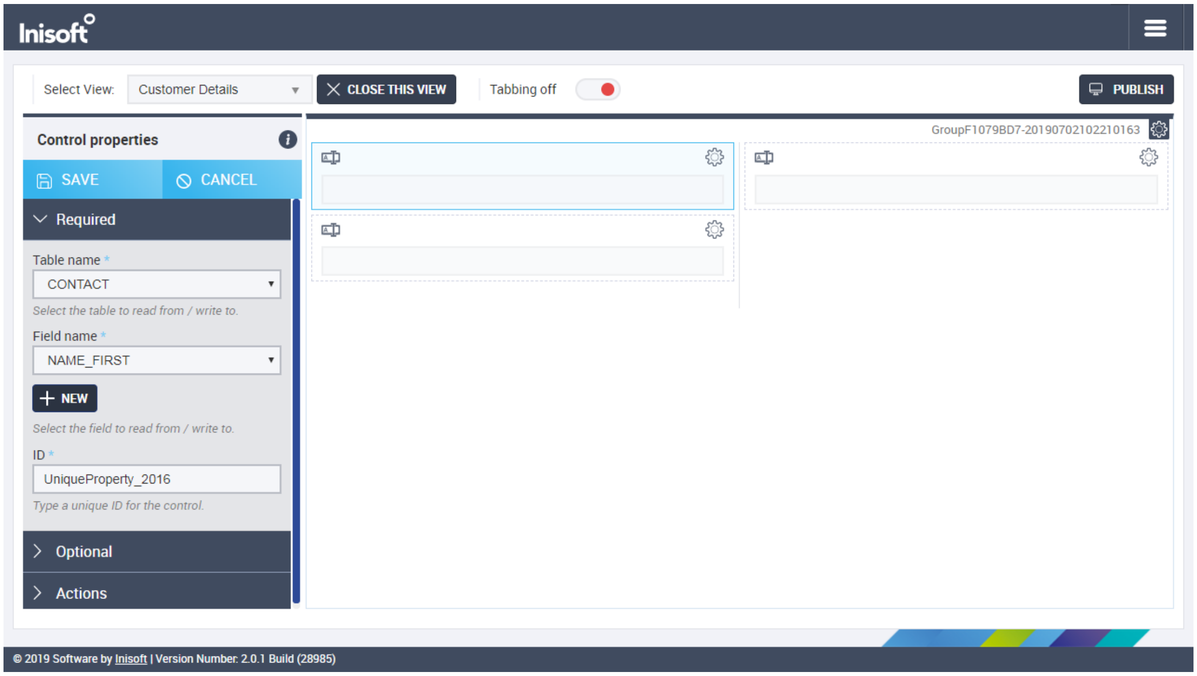
The environment that you're using with this tutorial has already been set up with a data universe called **Agent Desktop**, which has some database tables that you can use. You're going to link your three text boxes to fields in the CONTACT table.

> **Note:** If you're following this tutorial on an environment that has *not* been specially set up to support your learning, the universe you're using may not have a CONTACT table. Alternatively, it may have a CONTACT table but with different fields to those described below.
>
> You'll just have to proceed through the tutorial as best you can.

4.4 Editing the First Text Box

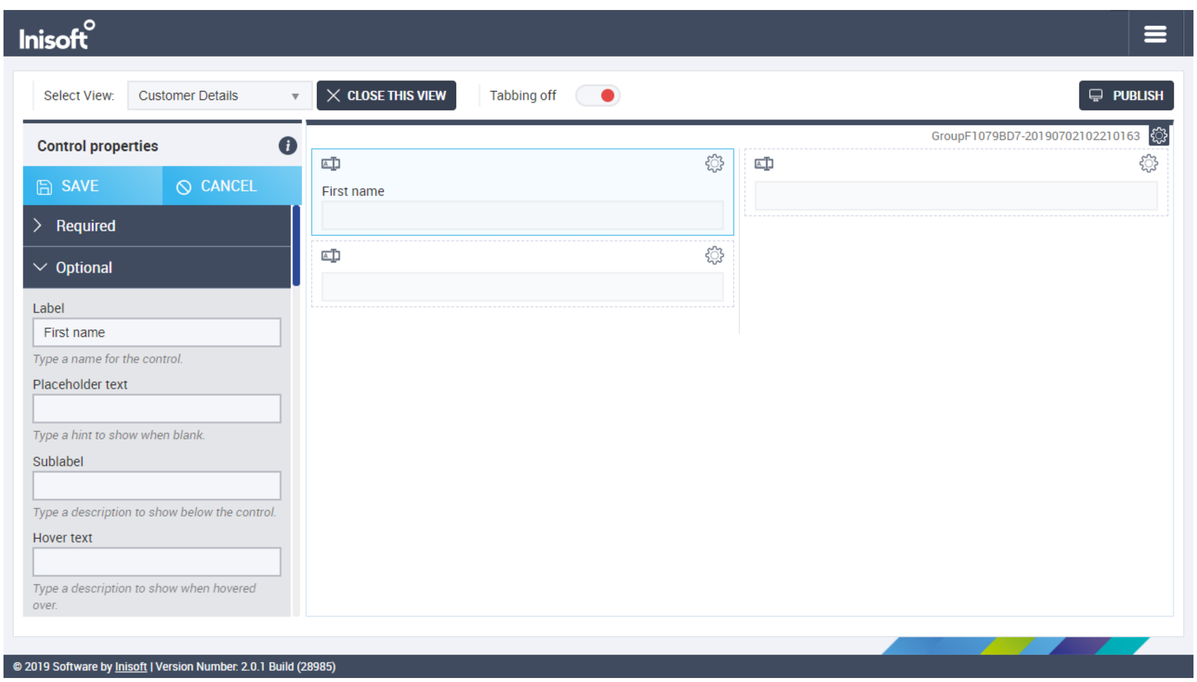
Back to where we were: the **Control properties** for the first text box.

In the **Table name** dropdown, select CONTACT. In the **Field name** dropdown, select NAME_FIRST. The **ID** field is automatically populated.



Next, click **Optional** to show additional properties.

In the **Label** box, type **First name**.



Last, click **SAVE** at the top of the **Control properties** menu.

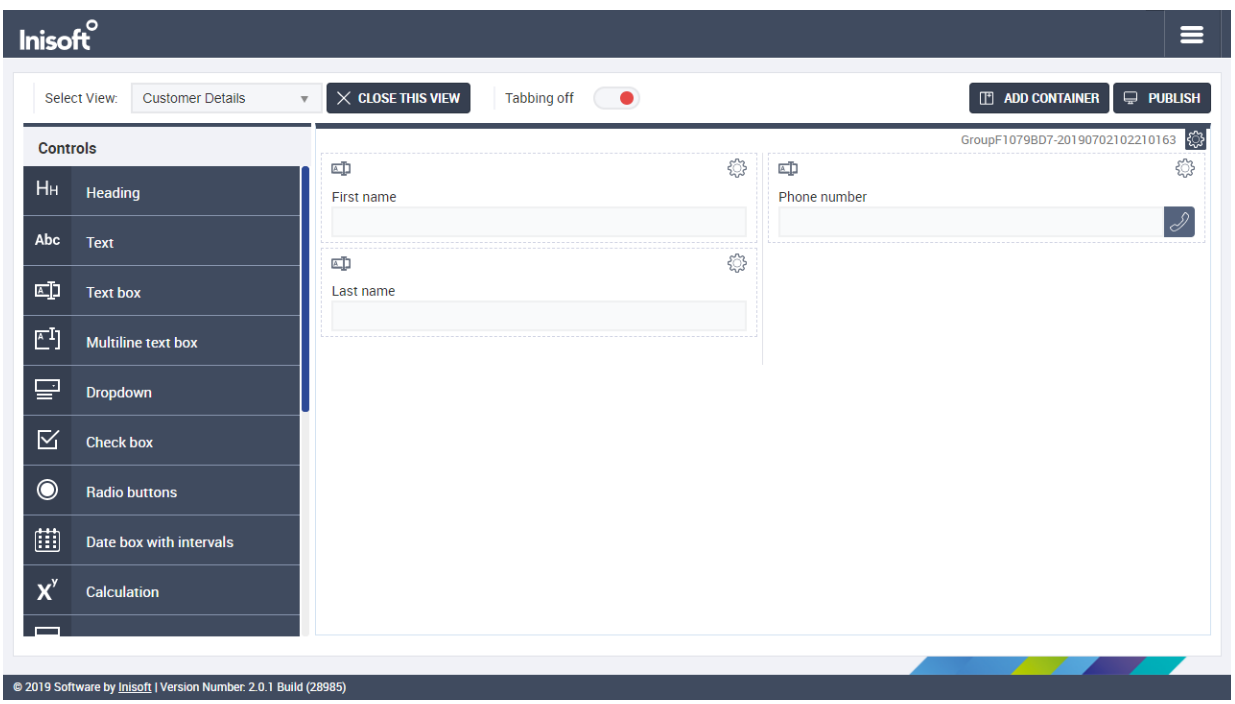
4.5 Editing the Next Two Text Boxes

Next, edit the other two text boxes using these details:

Table name	Field name	Label
CONTACT	NAME_LAST	Last name
INTERACTION_1	LKTL_NEXTDIALNUMBER	Phone number

In the optional properties for the phone number text box, set the **Text box type** to **PhoneNumber** and set **Dial on click?** to **Yes**.

When you're done, your desktop should look like this:



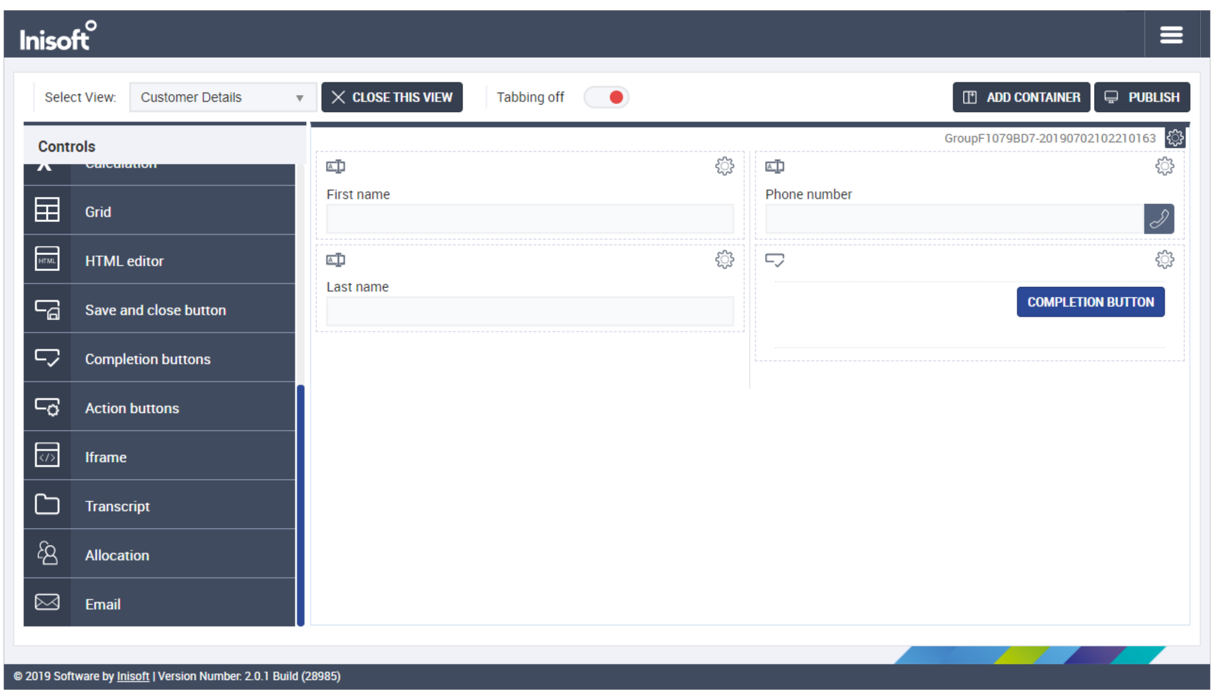
4.6 Adding a Completion Button

You've now been working in Designer for a while, so it'd be nice to take a look at your desktop as your agents will see it. This should help to put into context all the stuff that you've been doing.

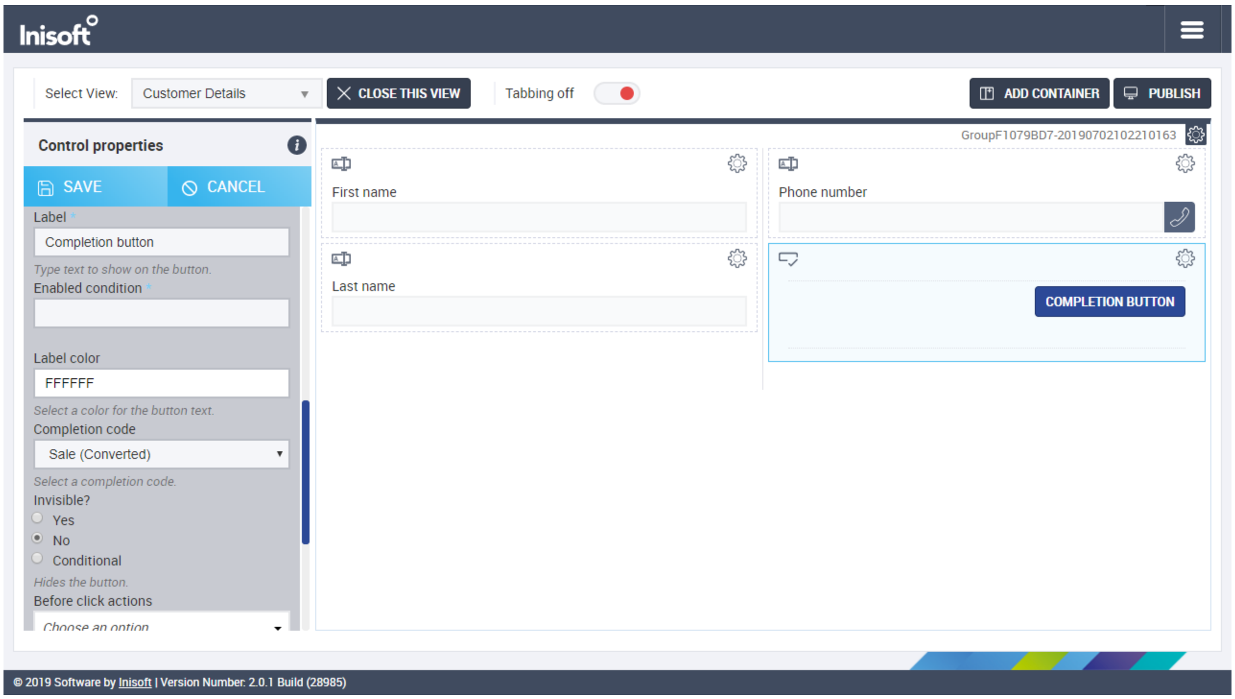
We'll do that soon, but first there's one more thing you need to add to your desktop: a completion button.

When an agent has finished speaking with a customer and has finished updating the customer's record, the agent clicks a **completion button**. This closes the customer's record and makes the agent available to be passed another call. You always need to include at least one completion button on an agent desktop.

OK, drag a completion buttons control under the phone number as so:



Just as with a text box, you can edit the properties of a completion buttons control. Do this, and edit the **Completion code** optional property to **Sale (Converted)**. Click **SAVE**.



Note: If you're following this tutorial on an environment that has *not* been specially set up to support your learning, the **Sale (Converted)** completion code may not be available. If that's the case, you'll just have to use a different completion code.

4.7 Closing the Data Entry Element

Next, click CLOSE THIS VIEW. The tab in your browser will close, and you'll see the Desktop Designer again.

> **Note:** If you're following this tutorial on an environment that has *not* been specially set up to support your learning, you'll need to take an extra step here.
>
> Drag a menu control onto your desktop, to the left-hand (25%) container. Edit the control's properties to give it a title like **Record creator**. Under **Items**, set **Item** to **Interaction record** and **Title** to **New record**.

In the Desktop Designer, click **SAVE** in the **Control properties** menu to close it, and then click 

to close the desktop too.

In the next chapter, we'll quickly go through updating your workzone. That's the last thing we need to do before we can test your desktop as your agents will see it.

5. Updating Your Workzone

All right, before you can test the simple desktop that you've created, we need to make some small changes to the configuration of your workzone.

5.1 What's a Workzone?

We encountered workzones once before, briefly, but we didn't explain them then, so let's talk about them now.

Think of a **workzone** as an area of your contact center business that agents can log in to. A travel company, say, might have two workzones: Personal Travel and Business Travel.

Some agents may only be trained in *either* personal travel *or* business travel. These agents would automatically be logged in to the appropriate workzone upon logging in to Syntelate XA.

Other agents may be trained in *both* personal travel *and* business travel. Such agents would be prompted to select a workzone upon logging in to Syntelate XA.

Often, as in our case, there's a one-to-one relationship between a workzone and a desktop. It *is* possible, though, to associate multiple workzones with the same desktop – but that's beyond the scope of this tutorial.

Also, it's at the level of the workzone that you configure such things as which CTI (Computer Telephony Integration) solution to use and whether the Worklist Engine should be used to pass records to agents. We'll configure these things now.

5.2 Updating the Workzone's Details

From the list of desktops, click **BACK TO TILES** and then click **Workzone Editor**.

Click for the workzone whose name matches the desktop that you created.

Next, follow these steps:

1. Change **CTI** to **Test telephony**.
2. For **CTI configuration (JSON)**, copy and paste the following:
   `{"AutoAnswerSeconds": "8","DefaultFromNumber": "+4400000000000"}`
3. For **Worklist enabled?**, select **Yes**.
4. Click **SAVE**.

What we're doing here is:

- Setting up the test telephony system, so that you can mimic making real calls.
- Activating the Worklist Engine for this workzone. The Worklist Engine is responsible for passing most types of interaction to agents (in our case, these interactions will be outbound calls for the agent to make).

Next let's look at testing your desktop.

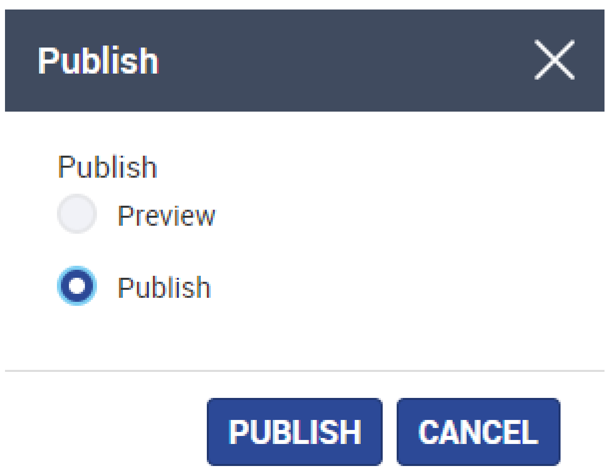
6. Testing Your Desktop

All right, so you've created a really simple desktop with a single data entry element. Now it's time to publish your desktop and test it out!

6.1 Publishing Your Desktop

Open your desktop again in the Desktop Designer. (From the list of desktops, click .)

Next, click . You'll be asked whether you want to preview or publish your desktop. Select **Publish**, and then click **PUBLISH**.



6.2 Viewing Your Desktop

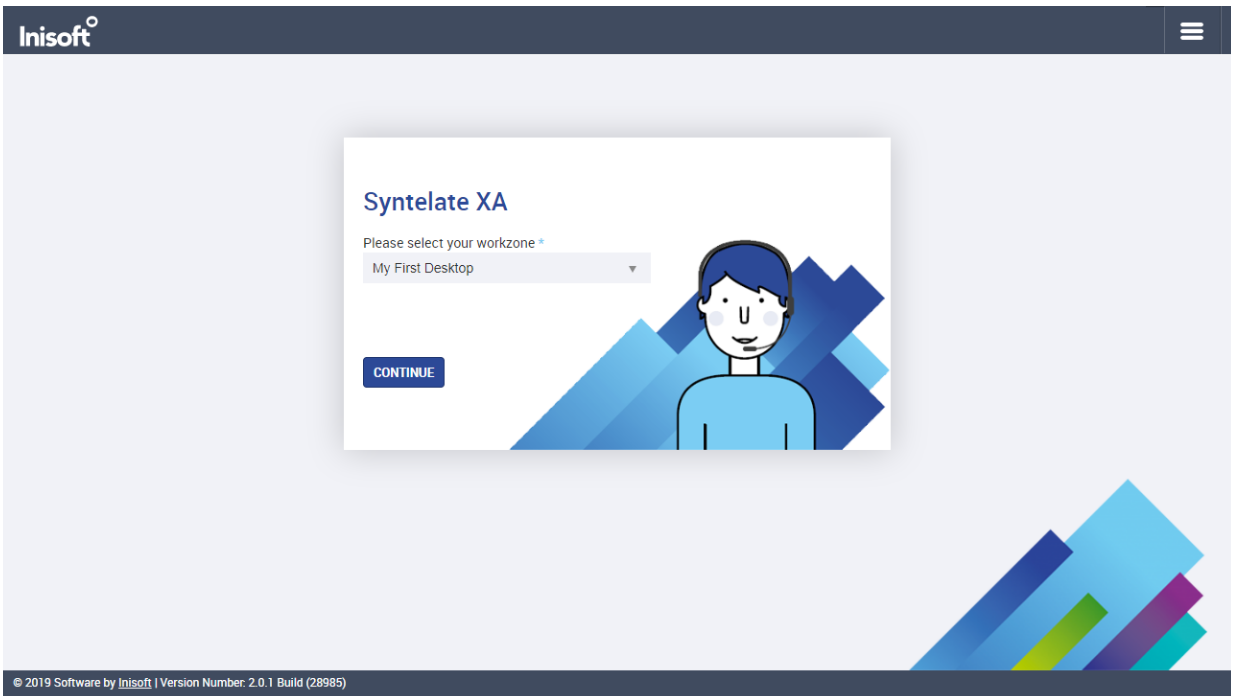
Now it's time to view your desktop as your agents will see it. So far we've been working in the Designer. When your agents use your desktop, that's called **runtime**.

In another tab or window of your browser, visit the URL for runtime.

6.3 Selecting Your Workzone

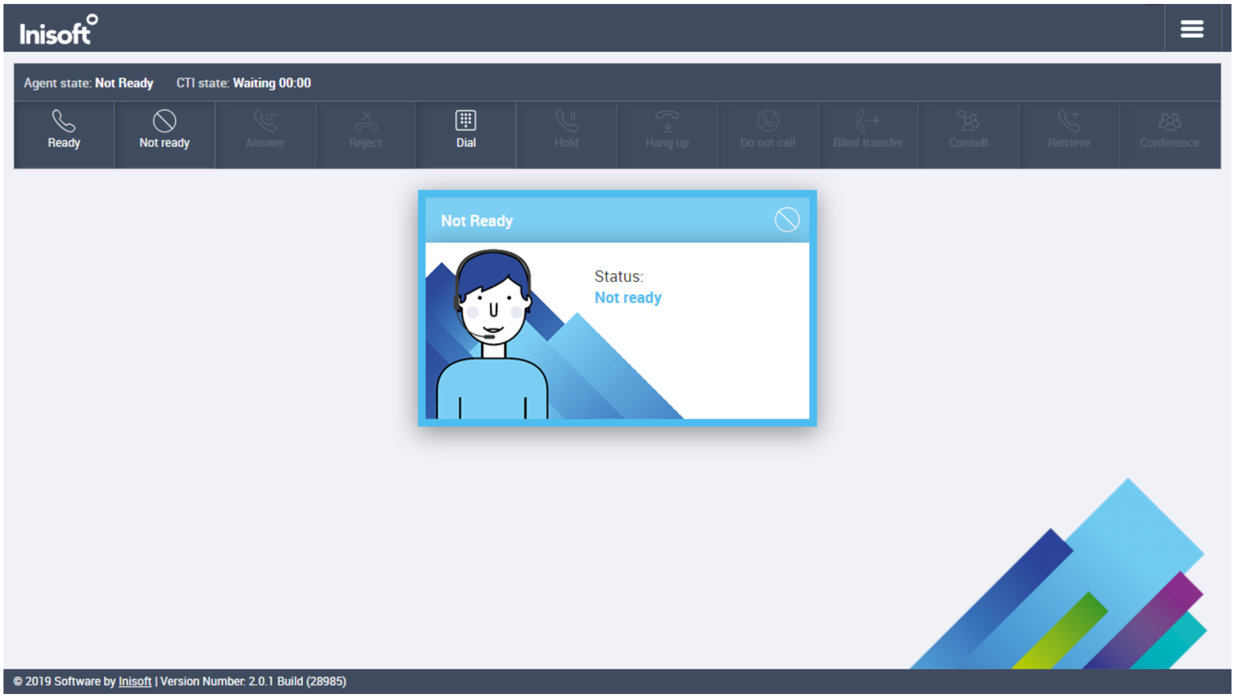
You'll be prompted to select a workzone.

Select the workzone that has the same name as your desktop, and then click **CONTINUE**.



6.4 Going Ready

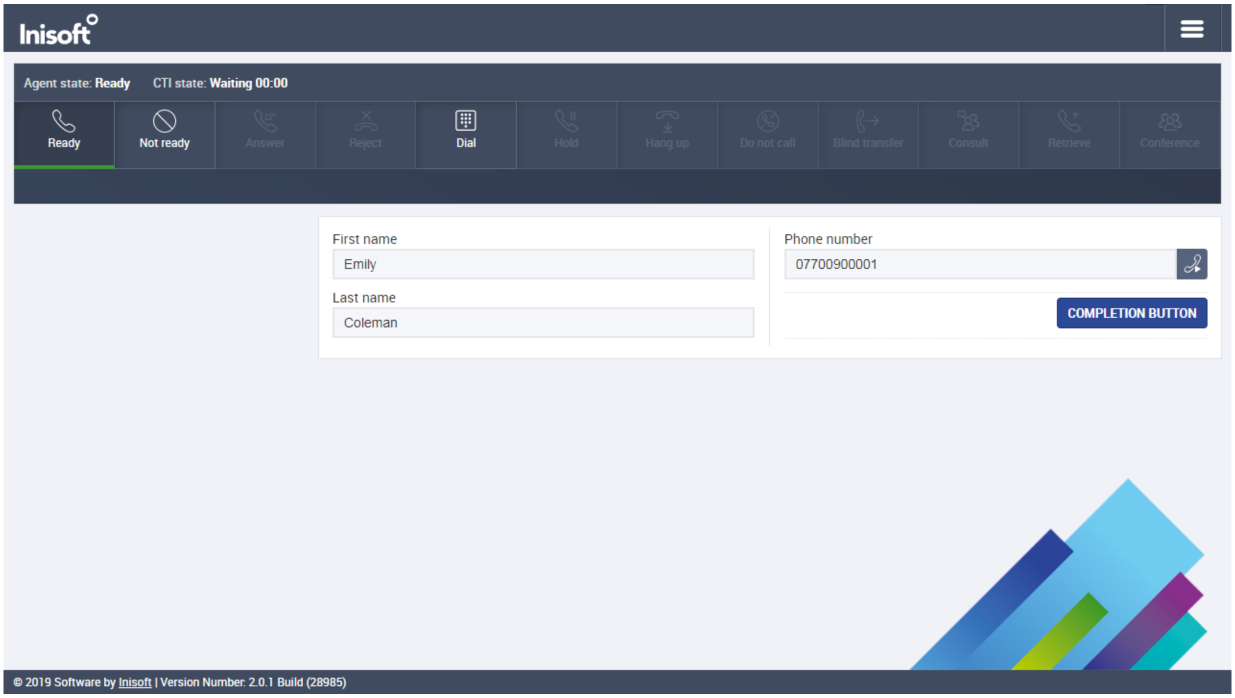
Once you've logged in to your workzone, you'll see the telephony toolbar and a not ready box. It's via the telephony toolbar that agents perform operations such as **answering** inbound calls and **hanging up** calls.



We set up your workzone to use the test telephony system. As such, you won't actually be able to make real calls to a number, but you can mimic dialing any number to get a feel for how Syntelate XA works.

As well as performing telephony operations, agents use the telephony toolbar to indicate their readiness to receive work.

For now, click **Ready** in the telephony toolbar. You'll be passed a dummy outbound call to make – and this will appear in the agent desktop that you've just created. Exciting!



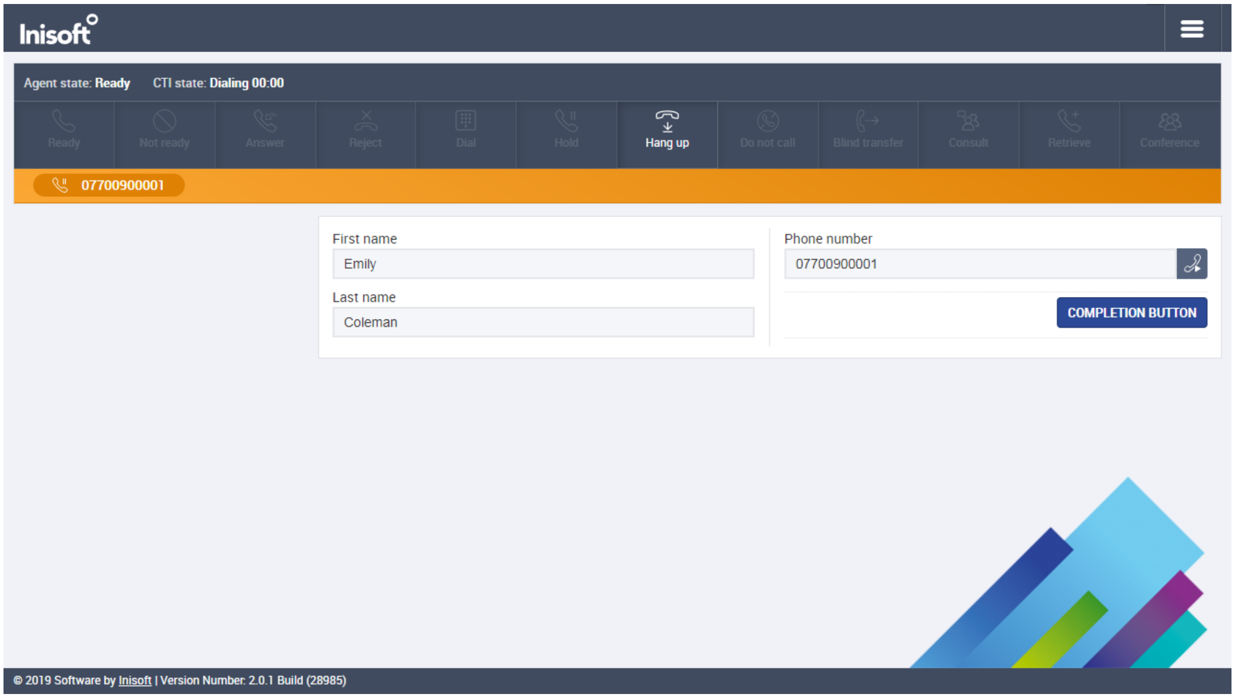
Note: If you're following this tutorial on an environment that has *not* been specially set up to support your learning, you probably won't be passed an outbound record at this point.

Instead click **New record** in the menu to manually create a new record. You'll then have to manually populate the three fields.

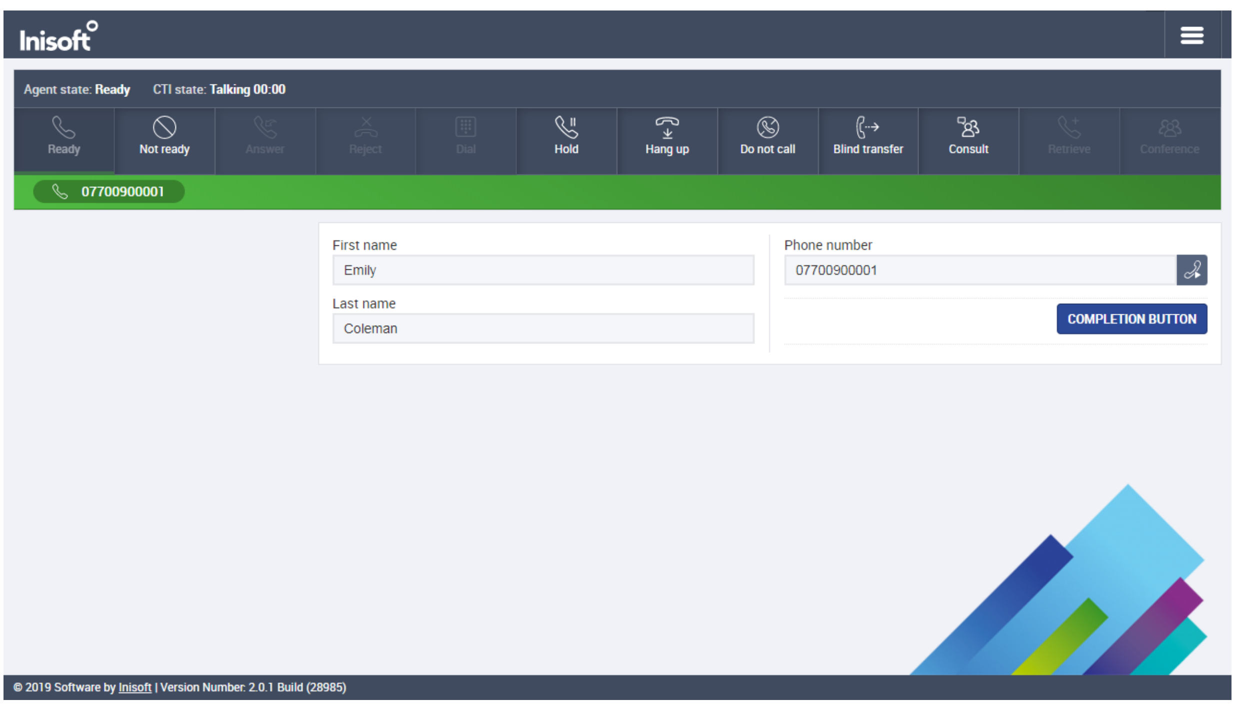
6.5 Making the Call

Click  beside the **Phone number** field to make the call.

The bar at the bottom of the telephony toolbar turns orange to mimic dialing.



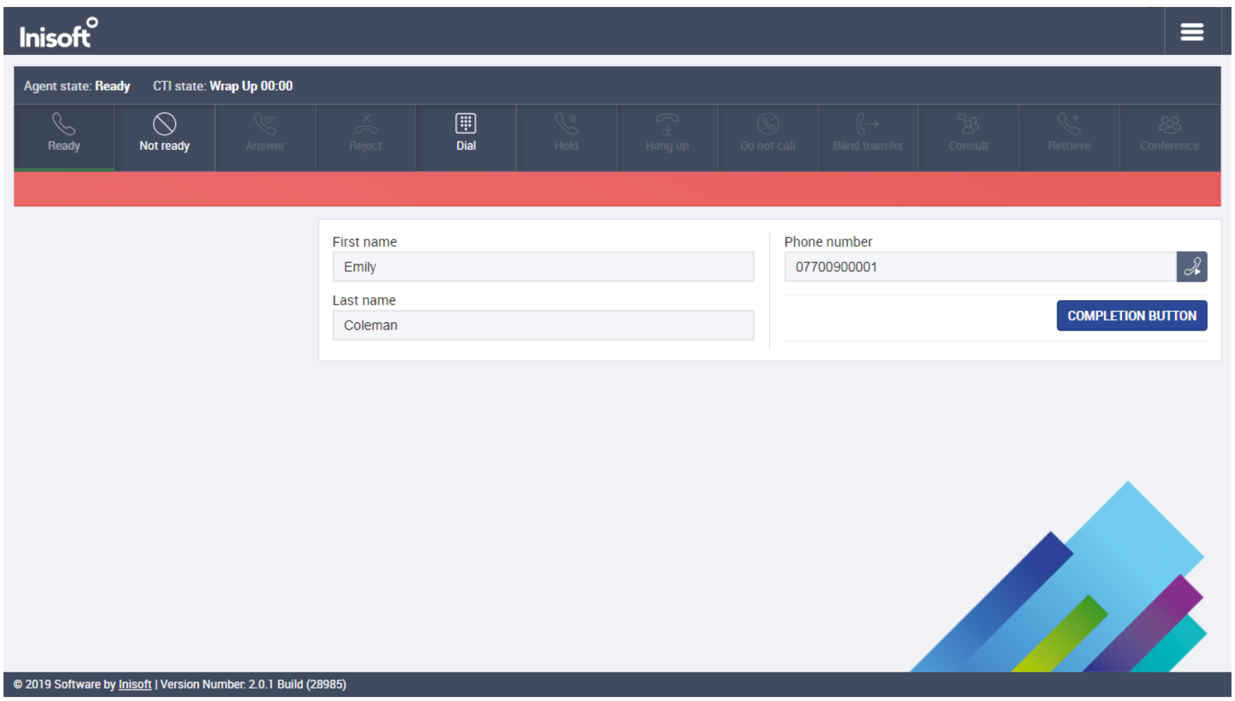
After 8 seconds the bar turns **green**, mimicking the customer having answered.



6.6 Completing the Call

Click the **Hang up** button in the telephony toolbar to hang up the customer.

When you do this, the telephony bar will turn red and you'll move into wrap-up. In this state, the agent can continue to update the customer's record before clicking the completion button to close the record.



Once you close the record, you'll be passed another outbound call to make. This time, rather than hanging up, click the completion button straightaway. This disconnects the customer *and* closes the customer's record, all in one go.

You can go through more records if you like, but with our rather simplistic desktop, there's not much else to see yet. We'll soon change that!

> **Note:** If you're following this tutorial on an environment that has *not* been specially set up to support your learning, you'll have to click **New record** whenever you want a new record.

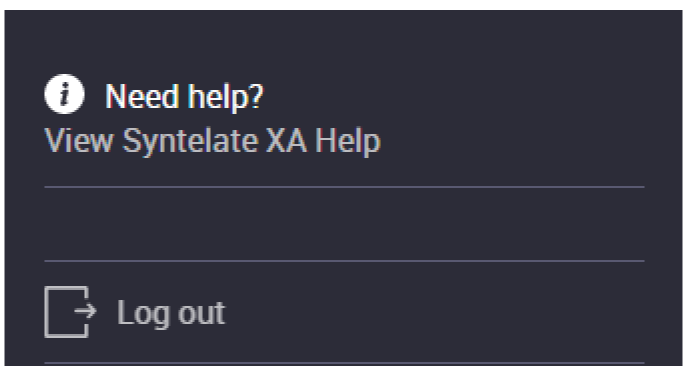
6.7 Going Not Ready

In the telephony toolbar, click **Not ready**. This tells Syntelate XA not to pass you any further records after you complete your current record.

Next, click the completion button to close your current record.

6.8 Logging Out

Click  to open the side menu:



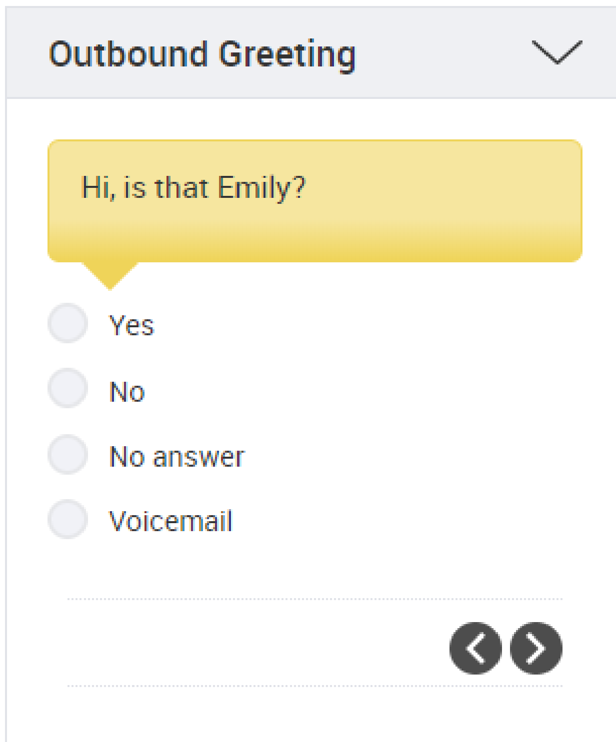
Next, click **Log out**.

Now it's time to go back to the Designer and make our desktop a little more interesting! In the next chapter, we'll look at designing a script.

7. Designing a Script

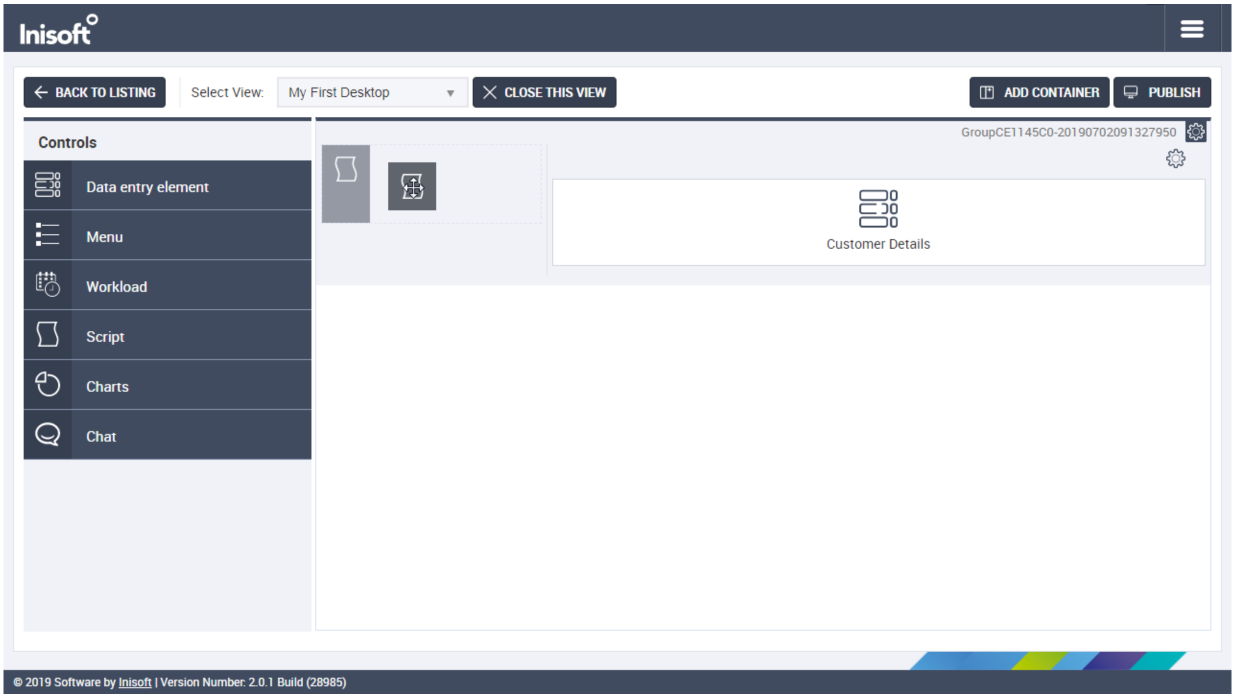
Let’s add a simple script to our desktop.

A **script** tells agents what to say and do during a call. It also defines the different paths that each call can go down.



7.1 Adding a Script

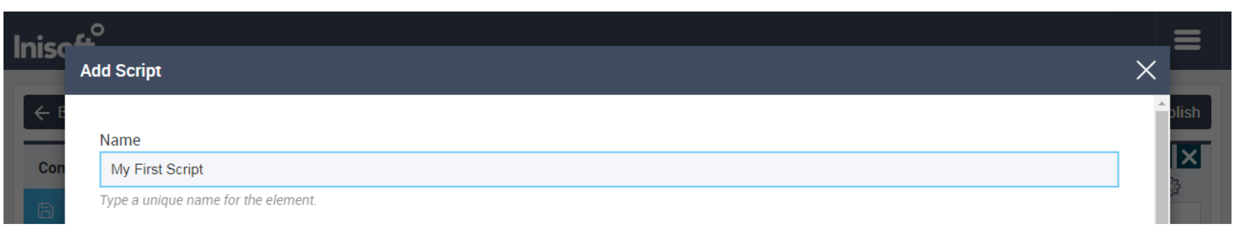
With your desktop open in the Desktop Designer, click and drag a script control to the left-hand column of the container.



7.2 Viewing the Script

Click the cogwheel on the script and select **Properties**.

Create a new script called something like **My First Script**, and choose to edit it now.



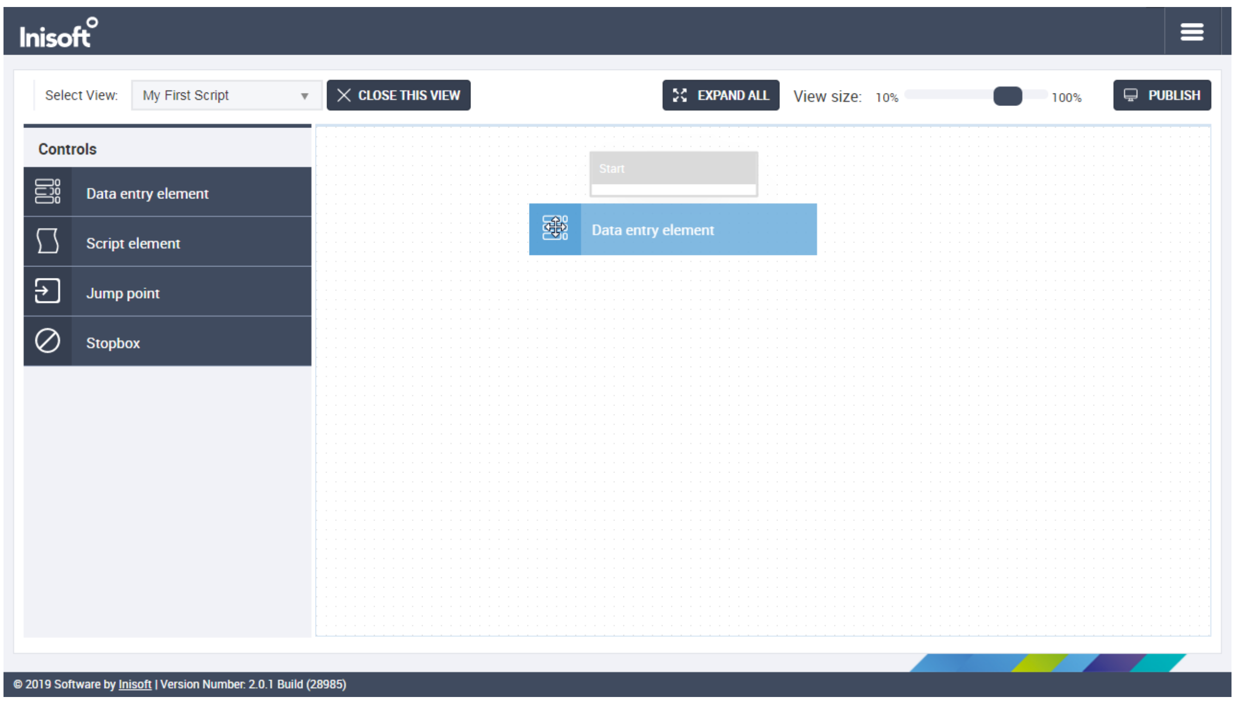
A new tab will open in your browser, and an empty script will show.

You've now jumped from the **Desktop Designer** to the **Script Designer**.

7.3 Creating Your First Script Box

You may remember, back when we first talked about data entry elements, we said that they had other uses than just letting agents view and enter customer data. One of those uses is in scripts!

Let's see that now. From the left-hand **Controls** menu, drag a data entry element to the **Start** box.



Click the cogwheel on this script box and select **Properties**.

Create a new data entry element called something like **Outbound Greeting**, and choose to edit it now.

You'll jump from the Script Designer to the Data Entry Designer.

Next, follow these instructions:

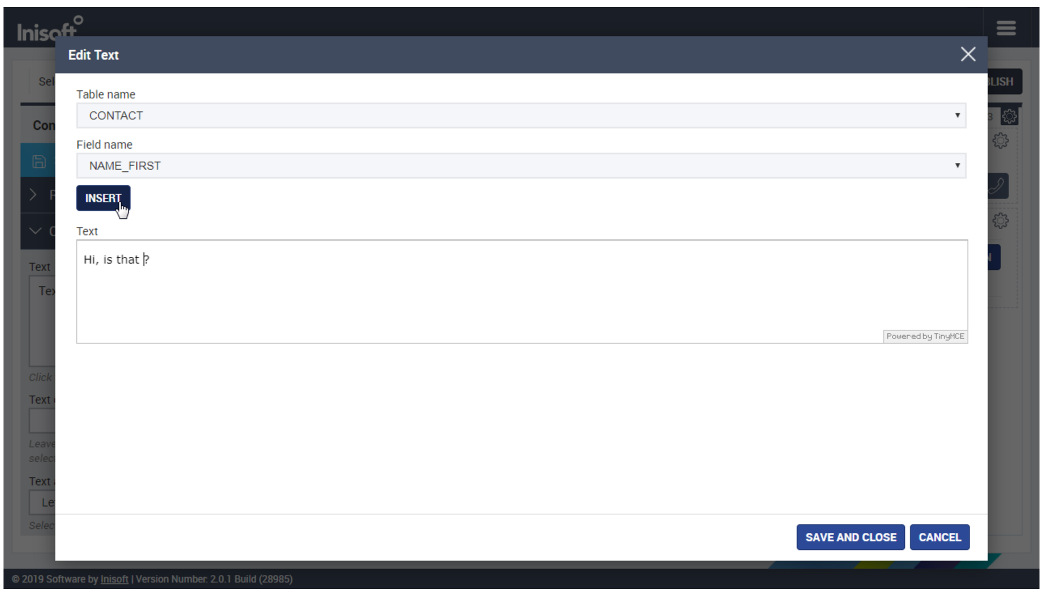
1. Add a 100% container to your data entry element.

2. Add a text control (*not* a text box control!) to this container.

3. Set the **Text** property of this control to: *Hi, is that [CONTACT.NAME_FIRST]?*

   At runtime, Syntelate XA will automatically insert the customer's first name.

Tip: Use the dropdowns to insert the reference to the customer's first name.



4. Set the **Text type** of this control to **Speech**. This indicates that the agent should read out these words as is.

5. Add a radio buttons control after the text control.

6. Associate the radio buttons control with the CALL_TOUCH field in the INTERACTION_1 table.

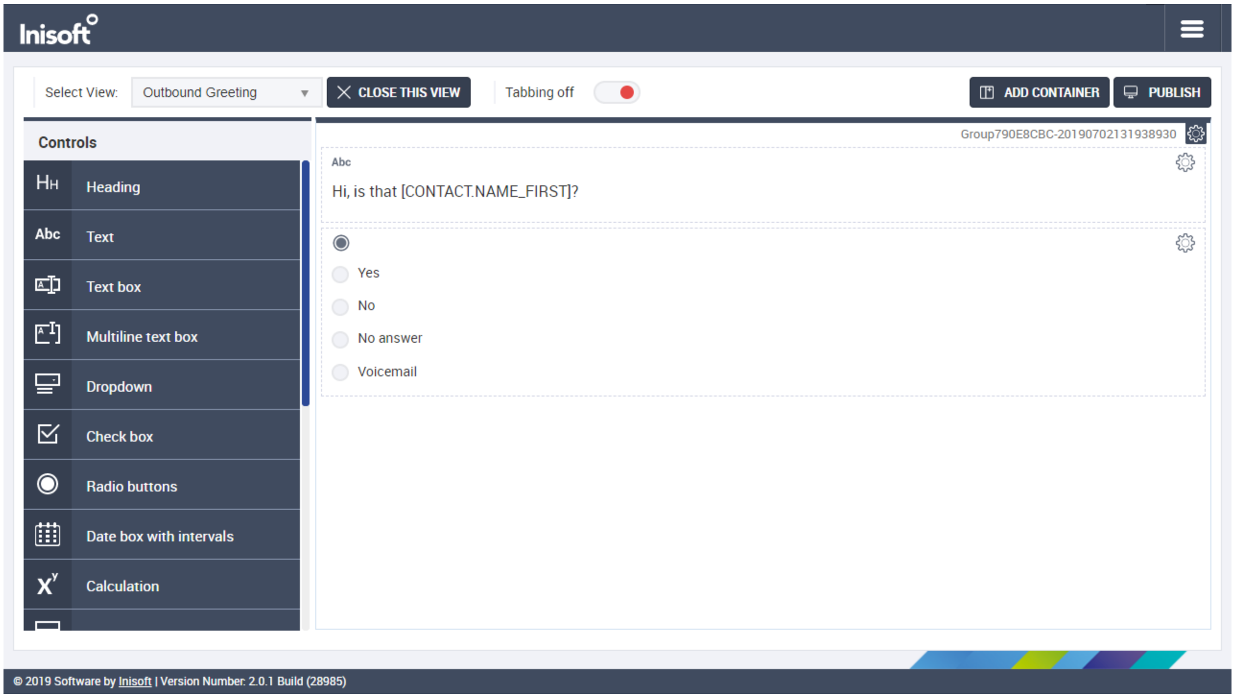
7. Edit the optional **Items** property of the radio buttons control so that it looks as so:

Value	Text
Y	Yes
N	No
NO_ANS	No answer
VMAIL	Voicemail

Tip: The **Text** field is the radio buttons text that the agent sees in Syntelate XA. The **Value** field is what is written to the CALL_TOUCH field in the INTERACTION_1 table.

You could use any values here, so long as they're distinct. In a moment we'll reference these values from our script.

When you're done, your data entry element should look like this:



Now click **CLOSE THIS VIEW** to return to your script.

7.4 Expanding the Script

In our first script box, we've added four possible options. Either:

- The agent has got through to the right person.
- The agent has *not* got through to the right person.
- The customer did not pick up.
- The call has gone through to voicemail.

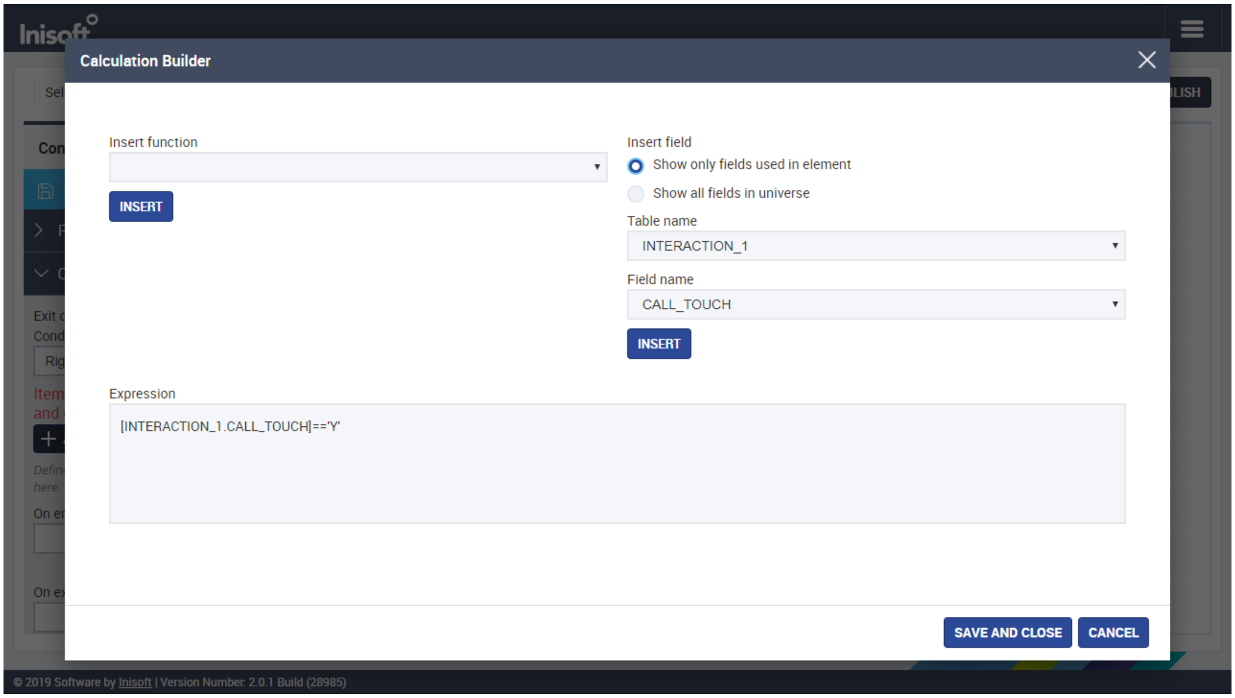
In our script, we now need to create branches for each of these four possibilities. This is done by defining **exit conditions**.

In the properties for the first script box, add four exit conditions as so:

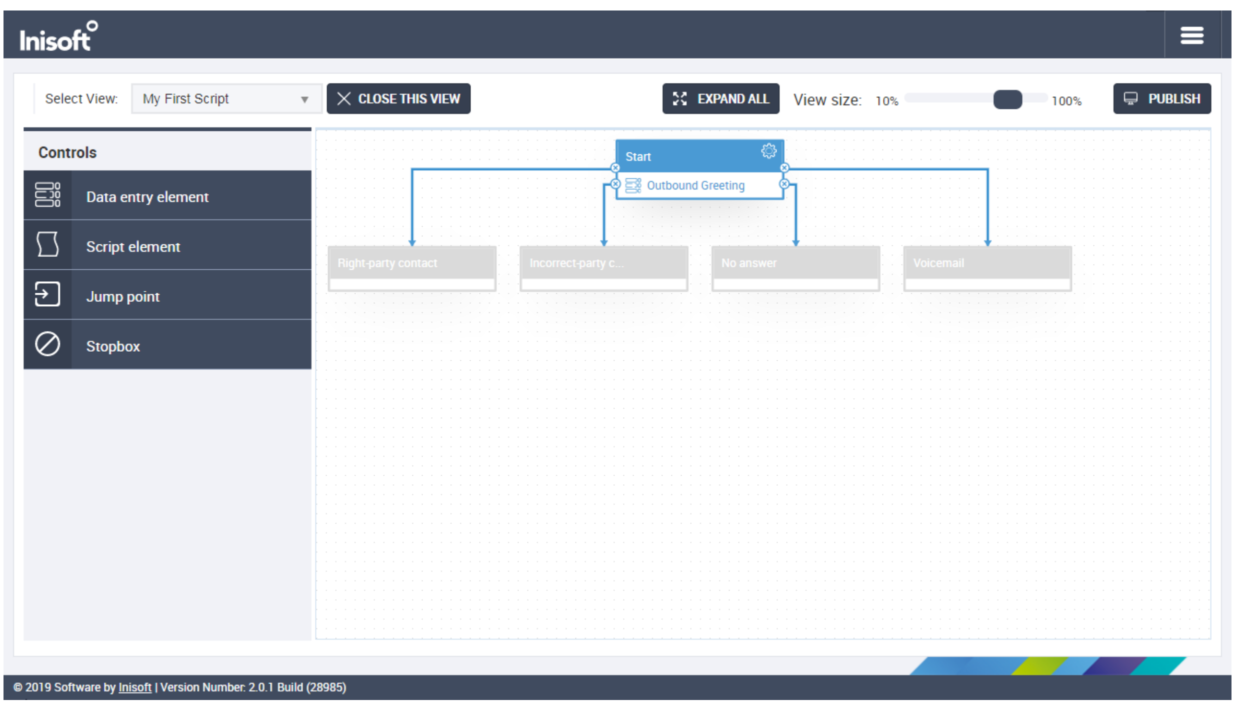
Condition name	Condition
Right-party contact	[INTERACTION_1.CALL_TOUCH] =='Y'
Incorrect-party contact	[INTERACTION_1.CALL_TOUCH] =='N'
No answer	[INTERACTION_1.CALL_TOUCH] =='NO_ANS'
Voicemail	[INTERACTION_1.CALL_TOUCH] =='VMAIL'

The **condition name** shows in your script in the Designer and helps you to easily follow what's going on. It won't show to agents, so you can really enter whatever is most helpful for you.

The **conditions** here match up with the radio buttons that you added to the data entry element. To add conditions, you'll use the **Calculation Builder**.



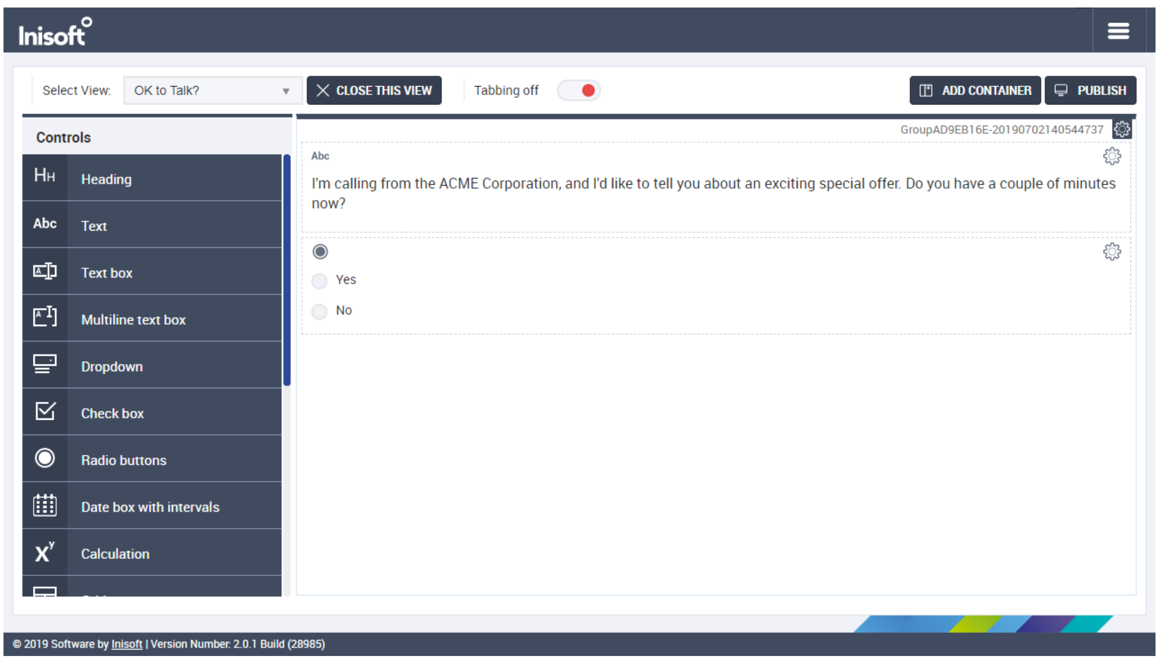
When you've added your four exit conditions, your script will look like this:



7.5 Further Expanding the Script

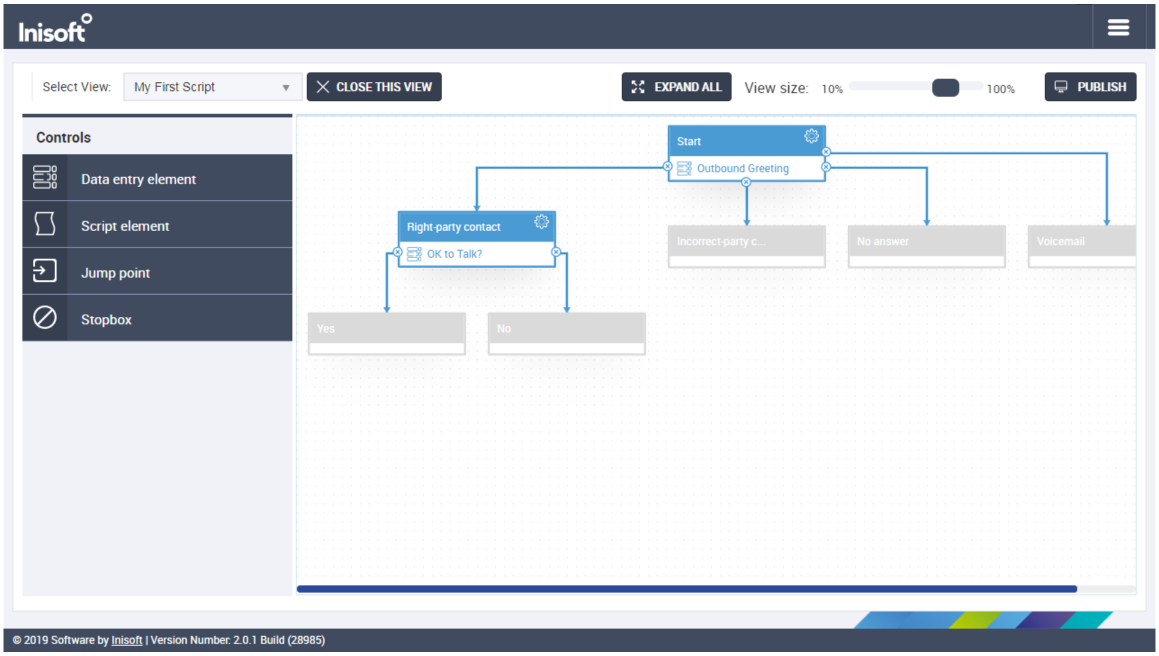
Hopefully you're starting to get a feel for how scripts works in Syntelate XA. Now let's work further on the right-party contact route.

Add a new data entry element that looks like this:



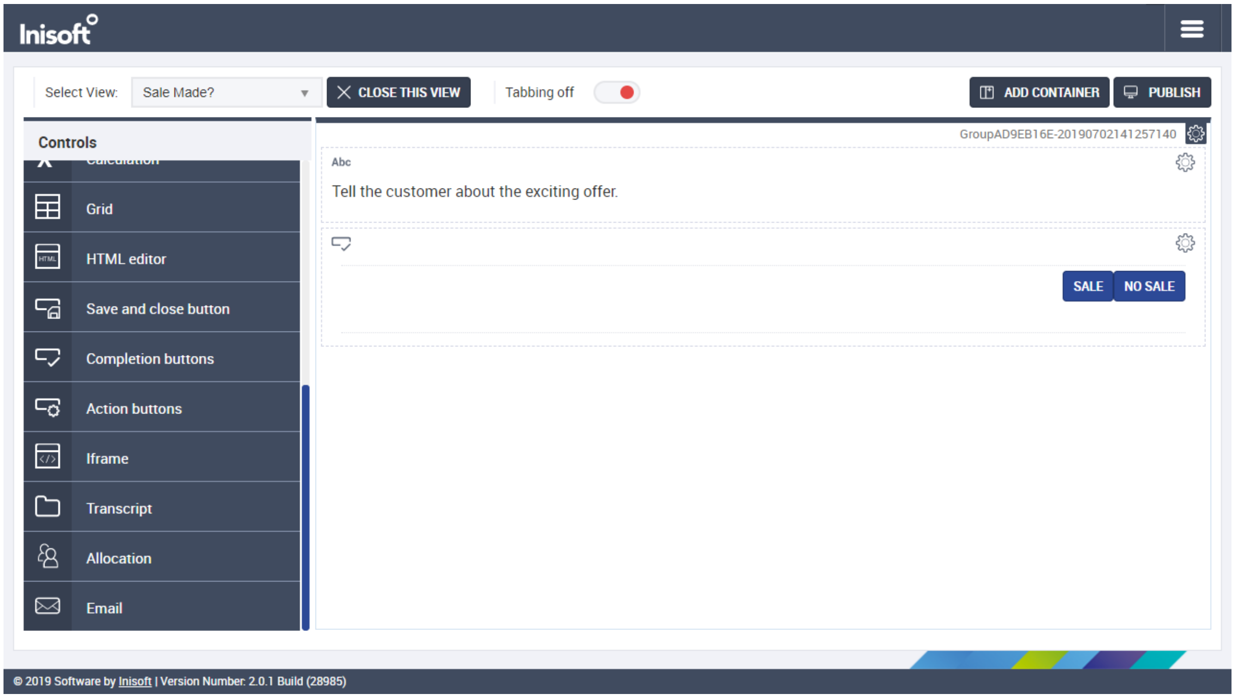
Link the radio buttons to field CALL_OK_TO_TALK in INTERACTION_1.

Add new exit conditions so that your script looks like this:



7.6 The Sale / No Sale Script Box

Add a new data entry element with two completion buttons that looks like this:



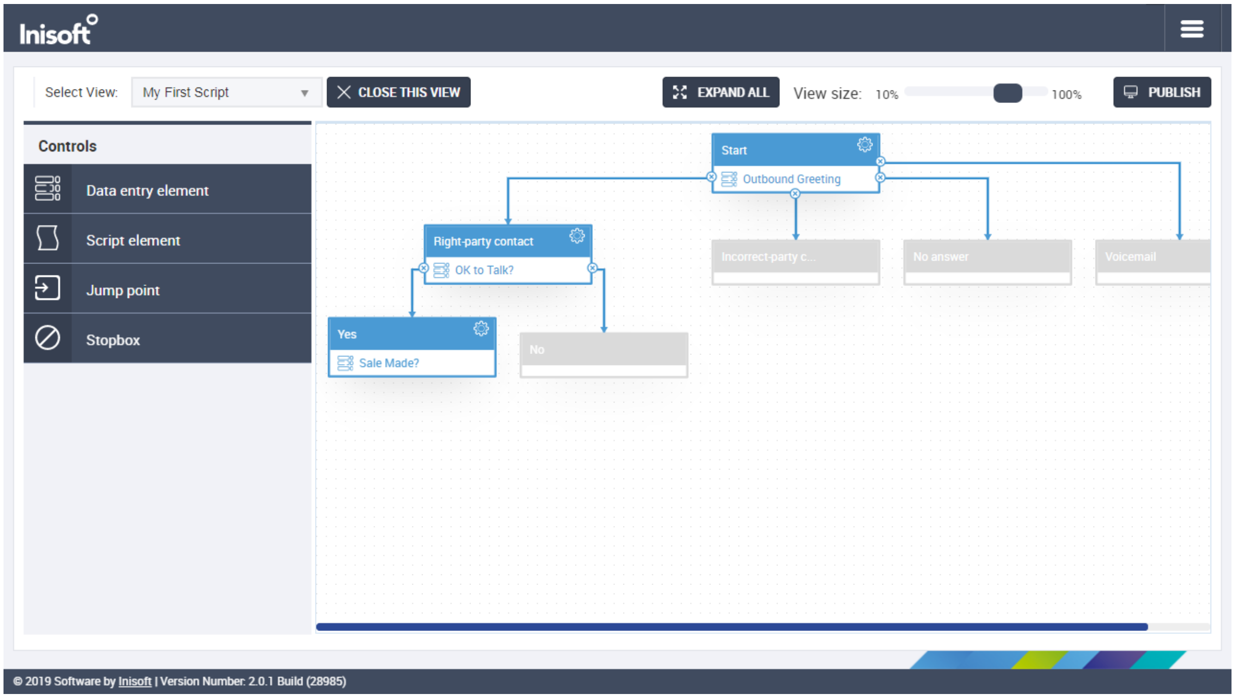
This time, for the text control, set the **Text type** property to **Instruction**. This indicates that the text is an instruction to the agent (rather than text to be read out to the customer).

Associate one completion button with the **Sale (Converted)** completion code and the other with the **No Sale** completion code.

Tip: Edit the **Label** property to change the text shown on a completion button.

To add a second completion button, click + ADD at the bottom of the **Control properties** menu.

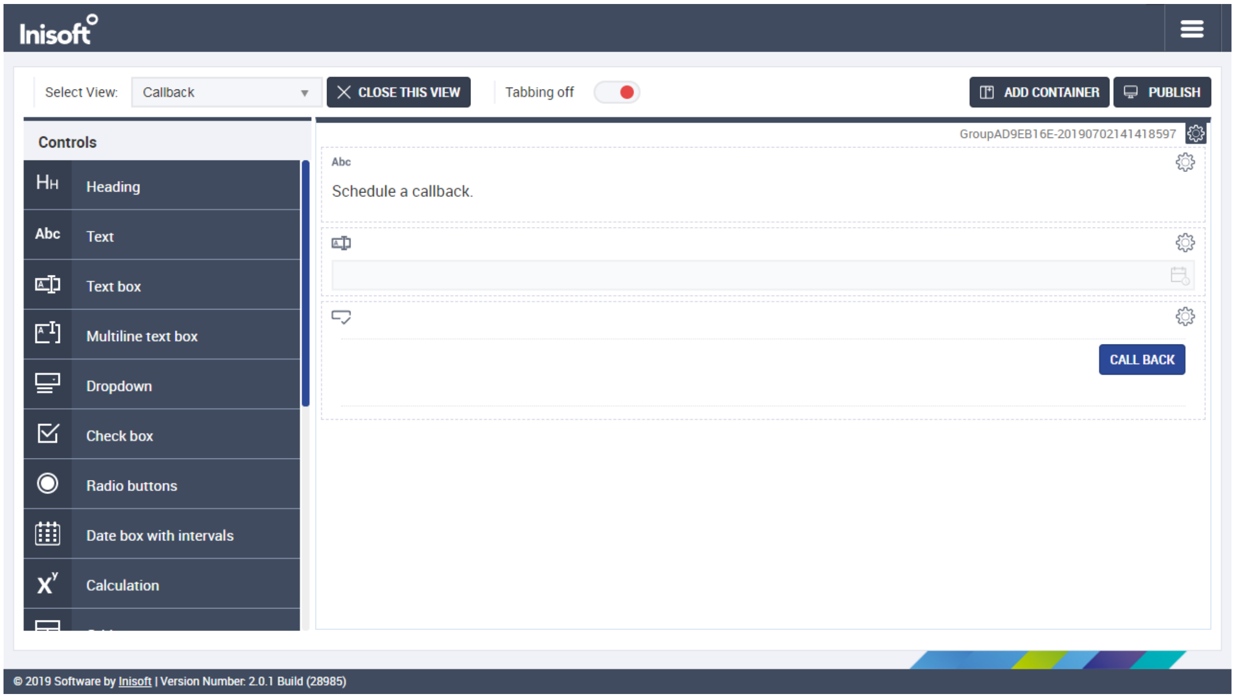
When you're done, your script should look like this:



7.7 Finishing the Script

All right, let's now quickly finish off our script. To cover all the other possible branches, we'll create a single data entry element that lets the agent schedule a callback.

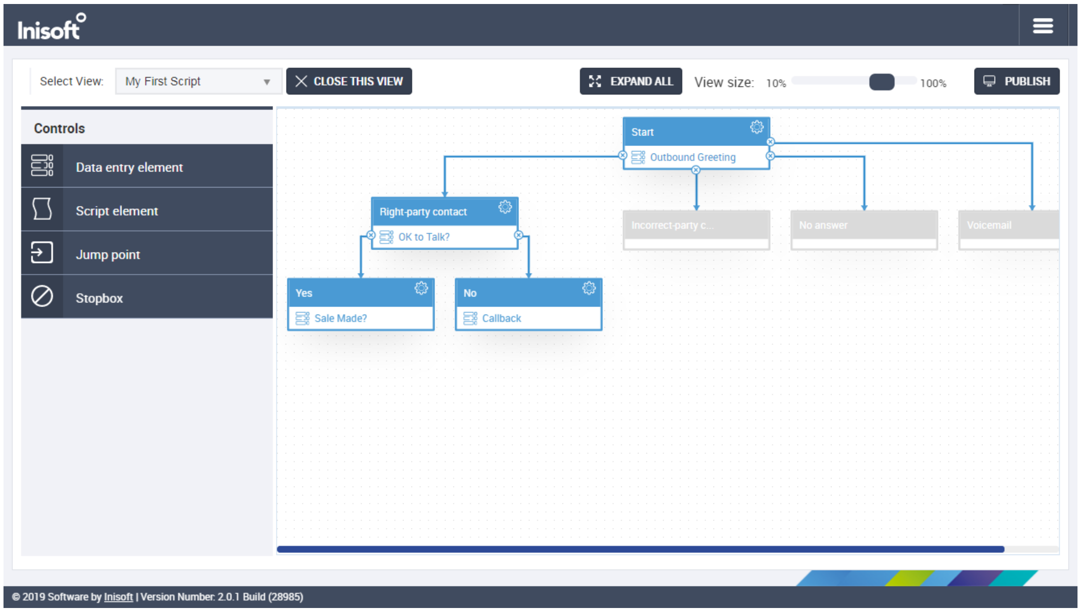
Drag this data entry element onto the **Right-party contact > No** box (although, for the record, you could really drag it onto any of the empty script boxes). The data entry element should look like this:



The middle control is a text box linked to field LKTL_RETRYAFTER in table INTERACTION_1. Set the **Text box type** property of this control to **DateTime**. This allows the agent to select a date and time via a date and time picker.

Set the **Completion code** property of the completion button to **Call Back**.

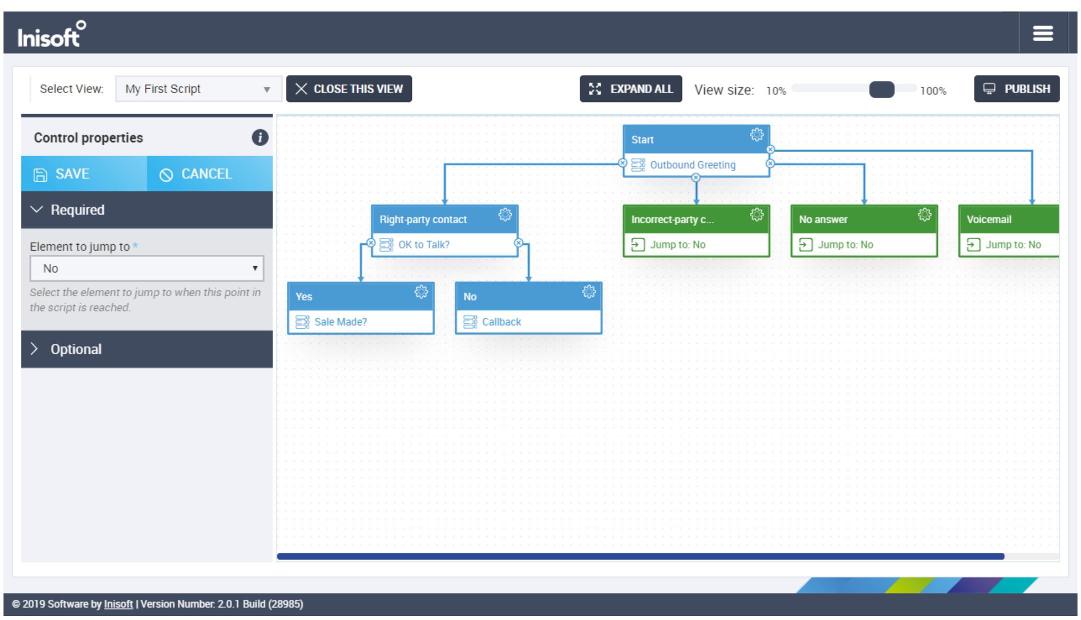
Your script will now look like this:



Now let's make use of **jump points**!

Drag jump point controls to each of the remaining empty script boxes. For each jump point, set the **Element to jump to** property to the **No** exit condition.

Your finished script should look like this:

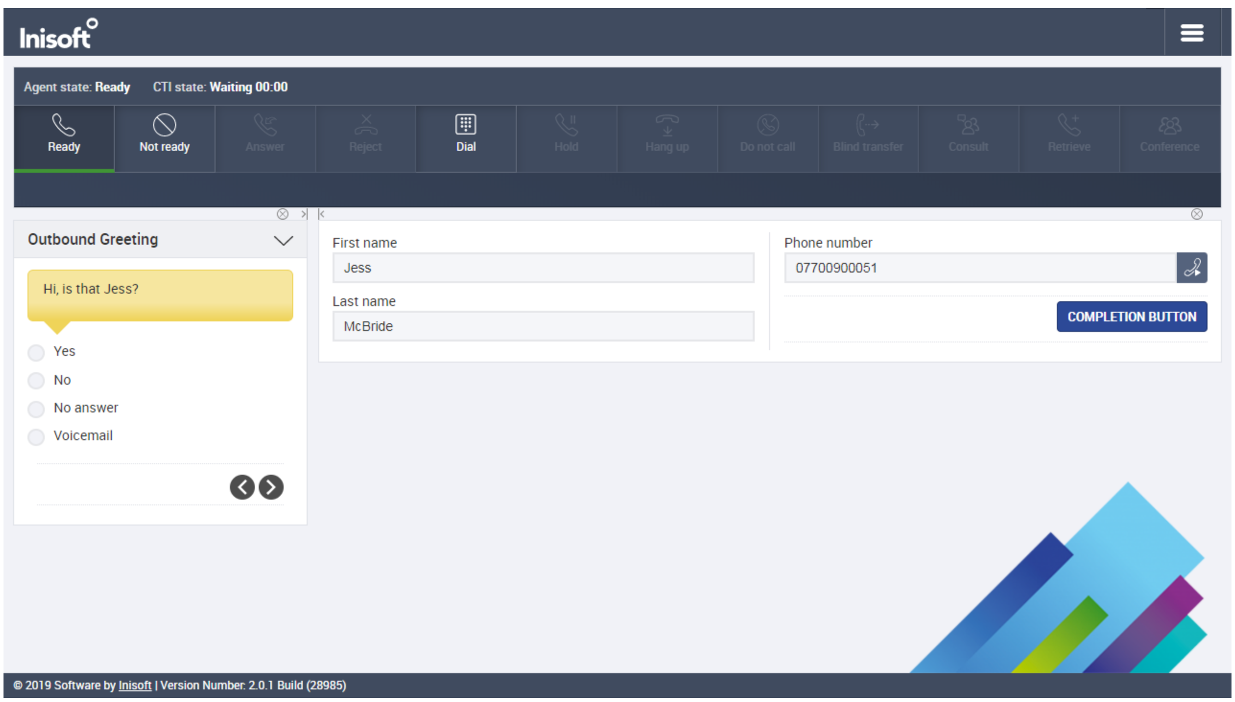


7.8 Testing the New Stuff

Phew!

Close your script and go back to the desktop. Publish it again to write the changes that you've made to runtime.

Now go back to runtime and test your script!



Try going through a number of different records, each time choosing a different path down the script.

Of course, this is a very simple script, but you should be getting an idea of what is possible in Syntelate XA.

In the next chapter, we'll finish up by looking at some other features.

8. Other Features

You've now created a really simple desktop with a script. By following the steps in this chapter, you'll get a quick look at some of Syntelate XA's other features.

This tutorial is by no means comprehensive, of course, and Syntelate XA has many more features than can be described here. But once you've finished this chapter, you should have a good understanding of the core concepts and be ready to explore Syntelate XA further yourself.

8.1 Removing the Completion Button

In the last chapter we added completion buttons to a script, but our original completion button is still there on the **Customer Details** data entry element. We don't need this anymore, so open this data entry element and delete it (click and then **Delete**).

By the way, you'll remember that in the last chapter we associated each of the three completion buttons with a completion code. This makes it easy to monitor the outcome of all calls. In our simple script, based on the completion button that the agent clicks, you can tell whether each call resulted in:

- A sale
- No sale
- A callback

In Syntelate XA, you could report on this in a dashboard, for example. Syntelate XA provides lots of features for reporting on the operation of your contact center – but that's not within the scope of this tutorial.

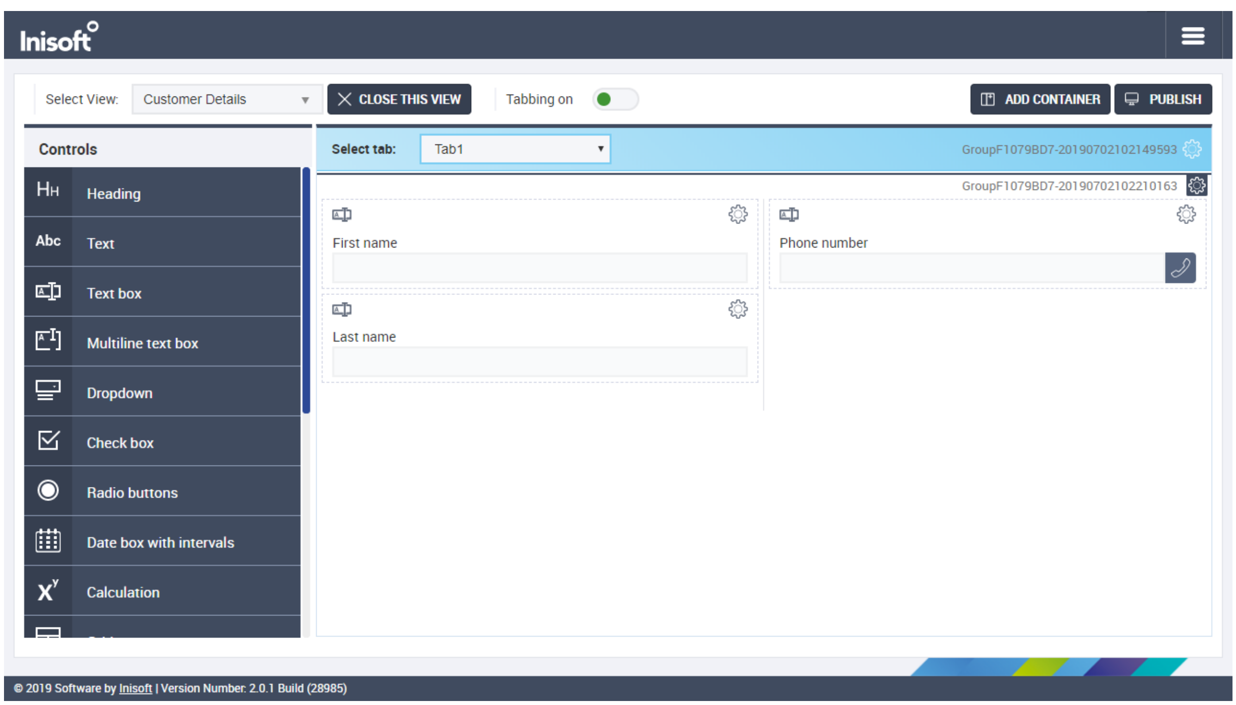
8.2 Tabbed Data Entry Element

A data entry element can have multiple tabs, each displaying different information.

Seeing as you've got the **Customer Details** data entry element open anyway, let's add a couple of new tabs to this.

First, click the tabbing toggle at the top to turn tabbing on:

Your data entry element should now look like this:



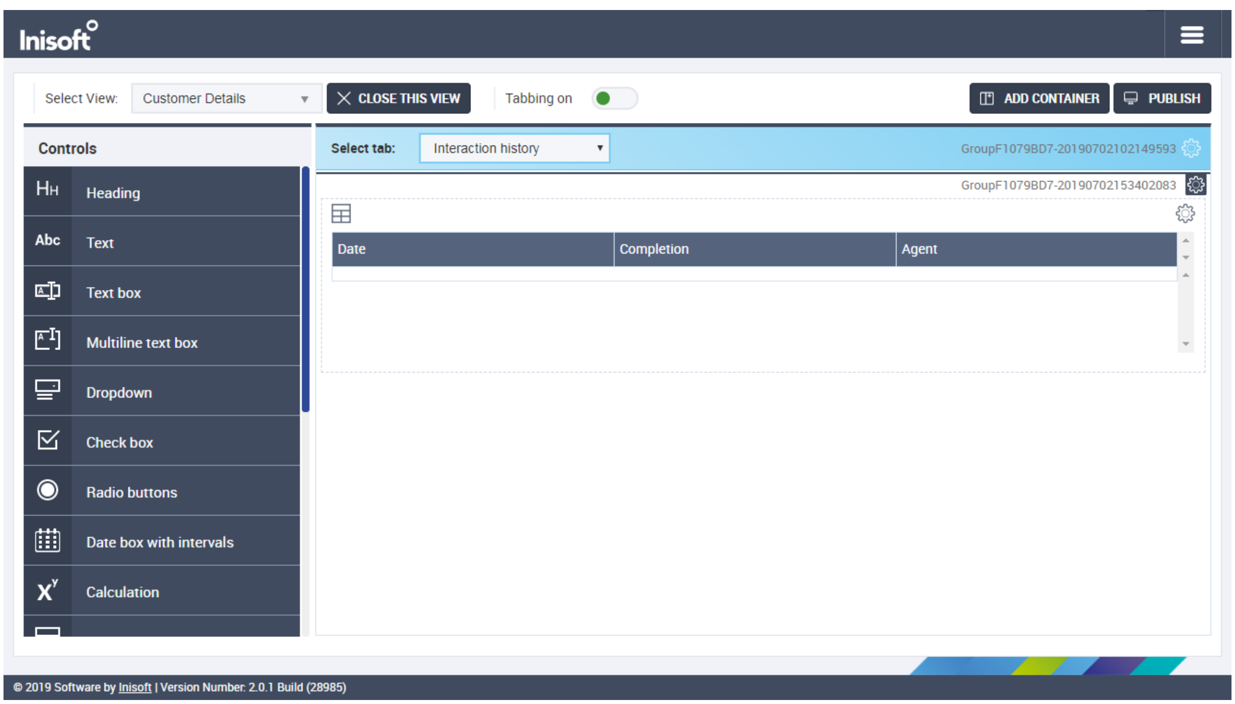
On the new blue bar, click to edit the tabs. Add two new tabs so that your list of tabs looks as follows:

Tab name	Order
Customer details	1
Interaction history	2
Sales promotion	3

8.3 Grids

On the **Interaction history** tab, add a grid control.

Rather than create a new grid, let's just use an existing grid: in the grid's properties, select **History of Current Interaction** as the **Grid name**.



At runtime, this will show a table (or “grid”) containing information about previous attempts to contact the customer. In your environment, this will be filled with nonsense data, but you’ll get the idea.

Anyway, let’s make some more changes before we test things out at runtime.

> **Note:** If you’re following this tutorial on an environment that has *not* been specially set up to support your learning, the **History of Current Interaction** grid may not be available to you.

8.4 Iframes

On the **Sales promotion** tab, add an iframe control. This lets you embed another web page into your agent desktop.

In a real environment, this would be handy where, say, your agents need to use another web application alongside Syntelate XA. Rather than your agents having to toggle between Syntelate XA and the other web application, you could simply embed the other application into Syntelate XA.

For now, though, let’s just embed a Wikipedia page in your agent desktop. This may not be something you’d ever do in a live environment, but at least it will give you a feel for how iframes work!

Edit the iframe’s properties and set the **Default source** property to https://en.wikipedia.org/wiki/Sales_promotion.

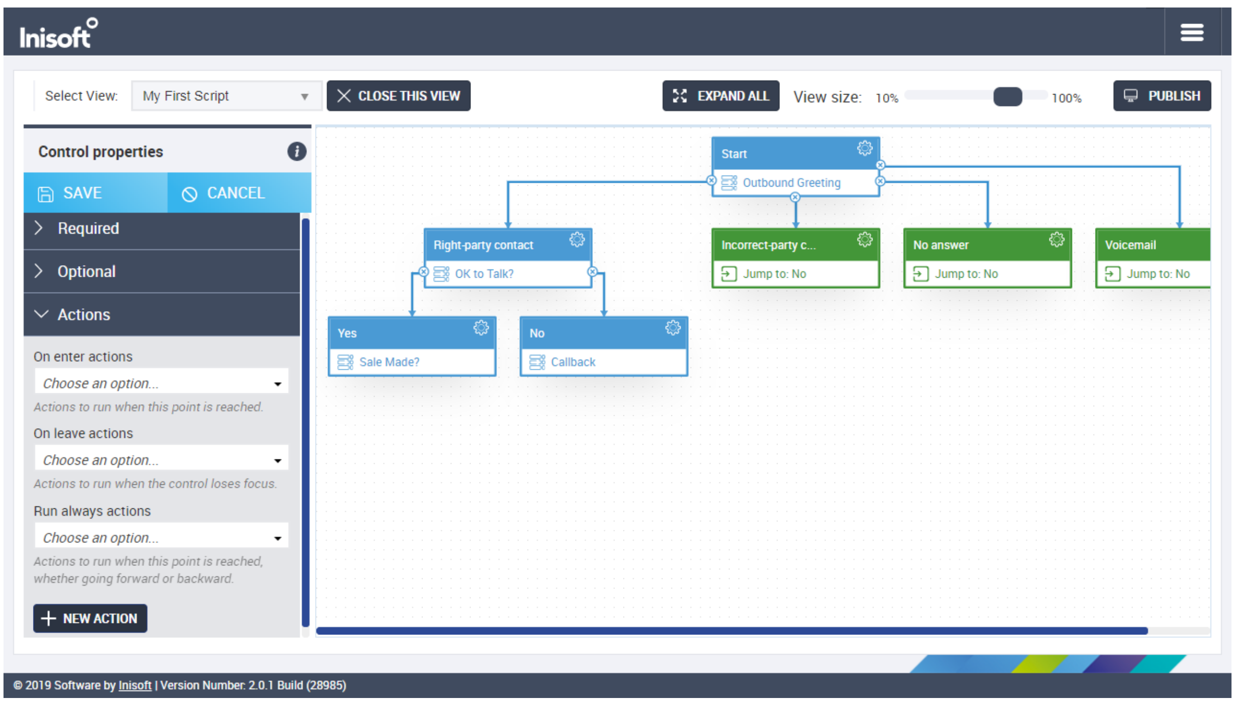
All right, that’s us finished in this data entry element, so you can close it.

8.5 Action to Activate Tab

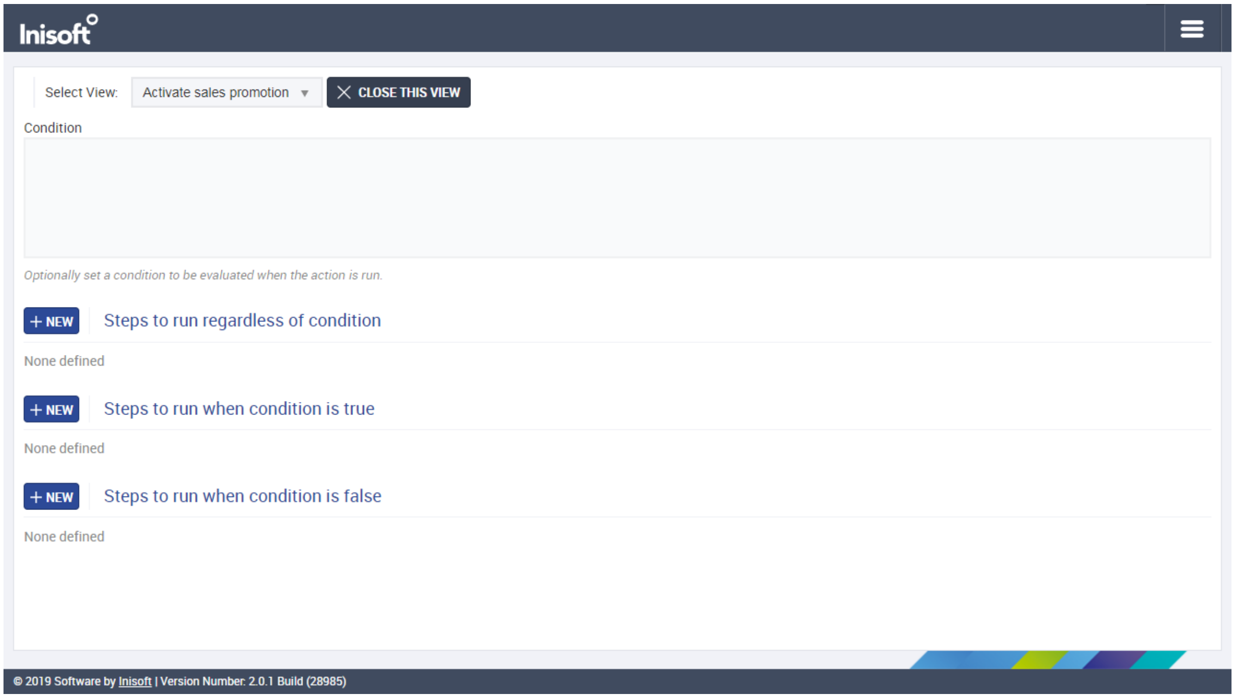
There's a really powerful feature of Syntelate XA that we've not mentioned yet. No introduction to Syntelate XA would be complete without a bit on actions, though, so we'll finish by adding a few actions to your desktop.

Open your script and edit the properties of the **Right-party contact > Yes** script box.

Expand the **Actions** property group, and click + NEW ACTION at the bottom.

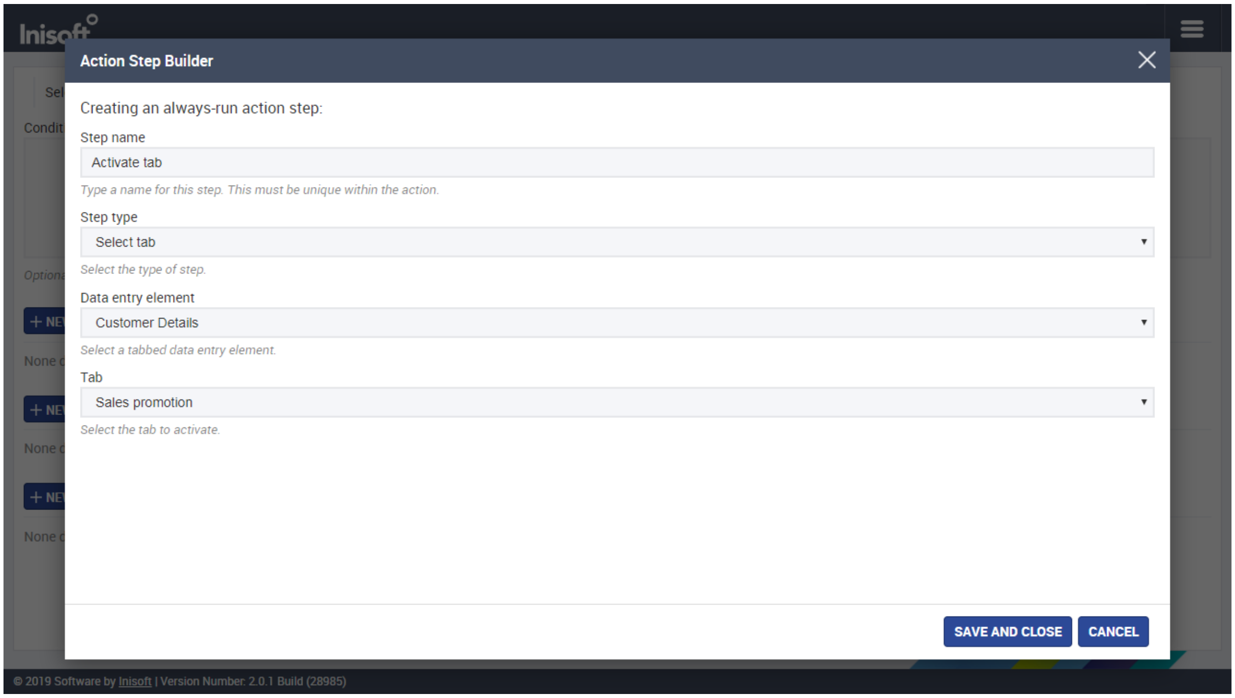


Call your action something like **Activate sales promotion tab**, and click **SAVE**. Choose to edit this now. You'll jump from the **Script Designer** into the **Action Builder**.



We'll talk about conditions later. For now, click **+ NEW** beside **Steps to run regardless of condition**.

Create a step like this:



Step name	Activate tab
Step type	Select tab
Data entry element	Customer Details
Tab	Sales promotion

Click **SAVE AND CLOSE**, and then click 

to close the Action Builder and jump back to the Script Designer.

Still in the properties of the **Right-party contact > Yes** script box, select your new action for **On enter actions**.

What does all this mean? It means that, at runtime, when your script reaches this script box, this action is run – and the **Sales promotion** tab is automatically activated.

Hopefully that makes sense, but if not, it will once you try it out at runtime.

8.6 Action to Validate Callback

Before we finish up, let's add a couple more actions. These are to tackle problems that you may have noticed in the last chapter, when you tried out your script.

Did you try setting a callback? If so, you may have noticed that it's possible to set a callback in the past. That's a bit silly, right? We should add some validation to prevent this, and we can do this via an action.

Open the **Callback** data entry element, and edit the properties of the completion button. Here, too, there is a + NEW ACTION button at the bottom. Click this, and create a new action called something like **Check callback is in future**.

We've not used the **Condition** box in the Action Builder yet – so let's give it a shot!

A **condition** is something that is evaluated when the action is run. You can perform action steps based on the result of this evaluation.

We want a condition that assesses whether the callback date and time is in the future and prevents the completion of the call if it's not. Here's how we'll go about doing that.

In the **Condition** box, enter: `[INTERACTION_1.LKTL_RETRYAFTER] > now()`

You can use the dropdowns to insert the database field and the function `now()`. This condition will evaluate as true when the LKTL_RETRYAFTER field is greater than the current date and time.

Next, add two condition-false steps as so:

Step name	Step type	Message
Show validation error	Show message	The callback must be in the future!
Cancel completion	Stop completion	N/A

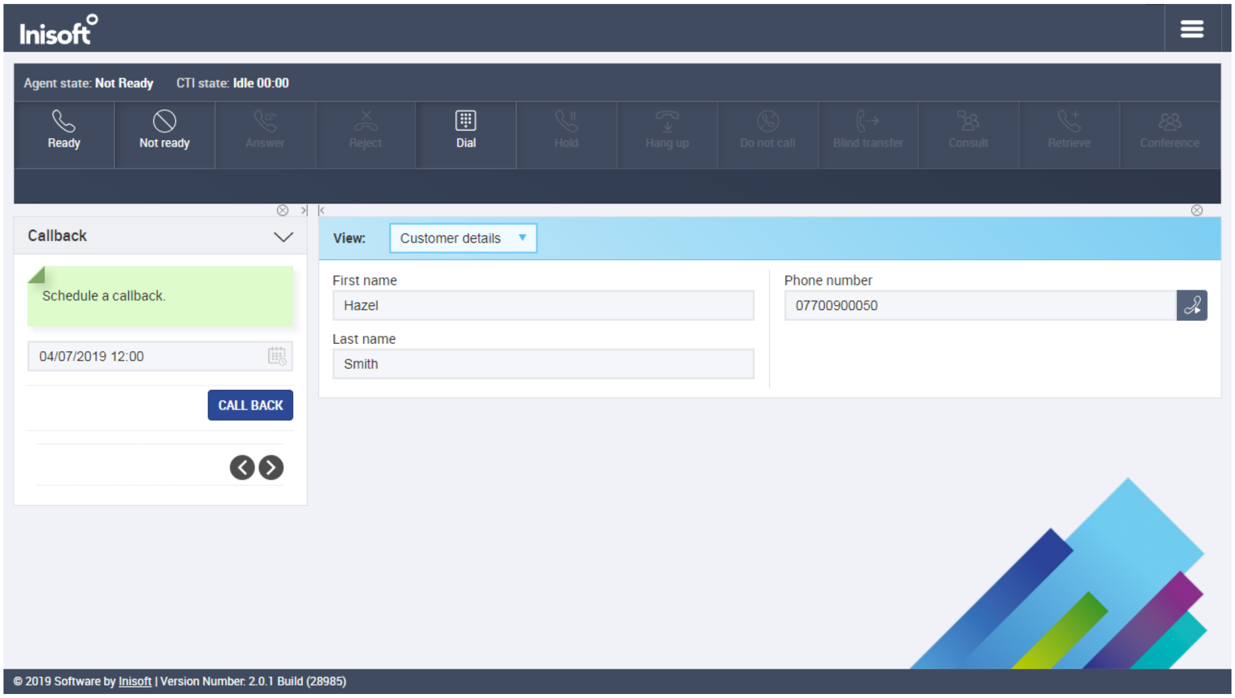
These action steps are run when the condition is false – in other words, when the LKTL_RETRYAFTER field is set to a date and time in the past.

Back in the Data Entry Designer, add your new action to the **Before click actions** property of the completion button. Any actions here run when the agent clicks the button but *before* Syntelate XA processes this. As such, this is where you should put any validation.

8.7 Expanding Our Validation Action

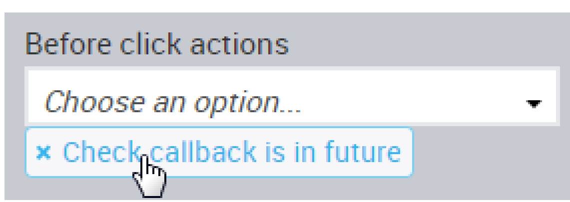
If at runtime you tried setting a callback for, say, a minute in the future, you may have found that at the callback time Syntelate XA didn't pass you the record that you expected but instead passed you other records to work on. We can fix this by increasing the priority of the callback record, so that Syntelate XA prioritizes it over any other records that are due.

If you *did* manage to get passed a callback record, you may have noticed that when the record came back to you, the script was still where it left off – at the callback page.



You probably don't want this. Instead, when an agent is passed a callback, you probably want the script to go back to the start. We can achieve this by clearing the fields that are set by the script.

OK, let's go back to our action. From the Data Entry Designer, you can get back into it by clicking the action name under **Before click actions**.



Back in the Action Builder, create some condition-true steps as follows:

Step name	Step type	Assignment statement
Increase priority	Assign value	[INTERACTION_1.LKTL_CBPRIORITY] =1
Clear right-party contact	Assign value	[INTERACTION_1.CALL_TOUCH] =''
Clear OK to talk	Assign value	[INTERACTION_1.CALL_OK_TO_TALK] =''

The first step increases the priority of the record to 1, the highest priority. The next two steps clear the fields that are set when an agent goes through the script.

Note: The second and third steps end with two single inverted commas (' and '), not a single double inverted comma ("). These steps set the fields to an empty value.

8.8 Runtime

Once you've finished your changes, go back to the desktop and publish it again.

Now test this new stuff at runtime:

1. Have a look at the interaction history grid.
2. Try going through the script to the **Sale Made?** script box, and see how the **Sales promotion** tab automatically activates. Take a look at the iframe.
3. Try setting a callback for a time in the past. You'll see a message telling you that the callback must be in the future. The completion operation will be canceled.
4. Try setting a callback for a minute in the future. When the callback record comes in, you'll notice that the script starts at the start again.

That's us now reached the end of this tutorial. In the last chapter, we'll consider where you should go from here.

9. What Next?

This has really just been the beginning of your journey with Syntelate XA. You've been introduced to the core concepts around agent desktop creation, and you should now have the confidence to explore Syntelate XA further yourself. Why not expand the basic desktop that you've just created with some new controls? There are a whole bunch of controls that you've not tried out yet.

You could also try exploring some of the properties that you've not used yet. For example, why not add a text box to the **Customer Details** data entry element with the **Required?** property set. This will prevent the agent from closing a record unless this text box has been populated – a simple form of validation that doesn't require actions.

Also, although you added a ready-made grid to your desktop, you didn't actually go into the Grid Designer itself, so you may like to take some time to explore that. Grids are really powerful. As well as creating grids that display information, you can create editable grids that let agents update fields and add new records.

Good luck! And remember, as you continue to explore Syntelate XA, if ever there's something you're not sure about, just look it up in the documentation.