Spoke Phone: Microsoft Power Automate to Dynamics365 integration

Getting started with the Spoke Phone example templates

Introduction

Power Automate is a service that helps you create automated workflows between your favorite apps and services, to synchronize files, get notifications, collect and update data, and more.

Power Automate is an ideal solution for customers using Microsoft, which provides all the infrastructure and system technology to integrate new and existing business workflows to Spoke Phone, and to take advantage of owning the integration and business workflows from end to end.

Spoke Phone has created a set of ‘boilerplate’ flows to get a customer up and running with the most common scenarios using MS Power Automate to integrate Spoke Phone to Dynamics365.

Microsoft Power Automate integrations into Dynamics 365 allows a Spoke Phone customer to:

Update the CRM within seconds, with key business intelligence by:

- Updating the CRM with new contacts when Spoke Phone users receive calls from unknown numbers, ensuring the CRM has a healthy view of all customers and their contact numbers.
- Updating the CRM with call logs keeping the CRM up to date with workforce activity.
- Automatically adding call notes, collected after calls end, to add any key insights or useful account information directly to the CRM.
- Adding Spoke Phone Call Disposition data, allowing the business to prompt the collection of important information to ensure follow up actions are automatically triggered.
- Adding Spoke Phone Market Insight response data, allowing the business to prompt the collection of Market Insight questions to capture and activate knowledge that is already known but not being utilized frequently

Keep your mobile workforce up to date by:

- Updating the Spoke Phonebook with new, updated, or deleted contact data.

Go even further with business workflows:

- Extend the integrations to support Dynamics 365 custom data fields, processes, or actions.
- Take ownership of the Power Automate Flows within their own environment and gain complete control over the business logic and behavior of the integrations from Spoke Phone to the Dynamics365 CRM.
- Extend the integrations to trigger data updates or workflows in other systems and create immediately actionable data from your mobile phone workforce.

Introduction 2

Getting Started 5
- Disclaimer 5
- Environment 5
- Optional Software: Postman 6

Power Automate Dynamics365 Templates Overview 7
- User Journeys 7
  - Overview 7

Using the Power Automate flows 8
- Installing the Power Automate flows. 8
  - Environment Considerations 8
  - Importing the Flows 8
- Testing the Power Automate Dynamics365 contact integrations 9
  - Set up the Spoke Phone API 9
  - Complete the Power Automate flow setup for Dynamics contact integrations 9
  - Build and load the Spoke phonebook 10
  - Test the CRM contact webhooks 11
    - Confirm setup 11
    - Test 1: Add new contact to Dynamics 365 11
    - Test 2: Edit a contact in Dynamics 365 11
    - Test 3: Add new contact to CRM 11
    - Test 4: Add new contact to CRM 12
    - Test 5: Merge two contacts in the CRM 12
  - Suggested Dynamics Customizations 13
- Testing the Power Automate Dynamics365 Spoke Phone event integrations 15
  - Complete the Power Automate flow setup for Spoke Webhooks 15
  - Set up the Spoke Phone Webhooks 17
  - Set up a Spoke Phone test form 18
  - Optional: Set up an excel log 19
    - Setup the template file 19
    - Configure the Power Automate flow 19
    - Set up the Spoke Phone Webhook for forms to Excel 20
  - Choosing the test method 21
  - Testing the Spoke Phone integrations with the Spoke Phone App 22
    - Preparation and device requirements 22
    - Running the tests with the Spoke Phone App: Recommended test script 22
      - Test 1: Add new contact to CRM 22
      - Test 2: Add unknown number to existing contact in CRM 24
      - Test 3: Add a call log for a known caller 25
      - Test 4: Add a note to a call 26
      - Test 5: Form submitted to a call 28
  - Alternative Test Method: Testing the Spoke webhook integrations with Postman 30
    - Getting started with Postman 30
    - Running the tests with Postman: Recommended test script 31
      - Test 1: Add new contact to CRM 31
      - Test 2: Add unknown number to existing contact in CRM 33
      - Test 3: Add a call log for a known caller 35
      - Test 4: Add a note to a call 37

Troubleshooting 40

Updating and customizing the flows. 41
- Probable extension cases: 41
- Other possible use cases: 41

Getting Started

Disclaimer

The Power Automate package is provided as an example set of templates to get you started with the most common integration scenarios from Spoke Phone to Dynamics365. We expect anyone using the provided templates to accept and own the operation and management of the Microsoft Power Automate example flows and ensure it is fit for purpose in the environment in which they are installed.

THE EXAMPLE FLOWS AND INFORMATION MADE AVAILABLE IN THE MICROSOFT POWER AUTOMATE SOLUTION AND THIS REFERENCE GUIDE ARE EXAMPLES ONLY AND ARE NOT TO BE USED AS IS FOR PRODUCTION PURPOSES. SPOKE NETWORK DISCLAIMS ANY WARRANTIES AND LIABILITIES UNDER ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, TORT, AND INDEMNIFICATION) IN CONNECTION WITH YOUR USE OF OR RELIANCE ON THE EXAMPLE FLOWS. ANY LIABILITIES THAT ARISE IN CONNECTION WITH YOUR USE OF THESE EXAMPLE FLOWS SHALL SOLELY BE BORNE BY YOU. BY ACCESSING AND DOWNLOADING THESE EXAMPLE FLOWS, YOU AGREE TO THE FOREGOING TERMS.

Environment

The following should be available to implement Power Automate:

- A Dynamics365 implementation, preferably with at least a sandbox and production environment.
    - A user able to customize Dynamics365 if required. Note: Customization is not essential, but improves the experience.
- An appropriate Microsoft Power Automate license with: access to use both premium connectors and custom data sources; having the capacity required to carry out the flows under the load required by your organization.
    - For more information please refer to the Microsoft Licensing requirements for your organization.
    - General Microsoft licensing information can be found here: https://docs.microsoft.com/en-us/power-platform/admin/powerapps-flow-licensing-faq
- **Two mobile phones,** one with a number not known in your sandbox CRM, **another phone** with Spoke Phone installed.
    - Alternatively, Postman can be used to simulate phone call events, although we recommend using phones as a preference.

Optional Software: Postman

NOTE: If you have not yet rolled out Spoke Phone to your end-users, we suggest testing with the live Spoke Phone App and a sandbox dynamics environment, rather than using Postman to simulate Spoke Phone webhooks.

When installing and testing the integrations, ***especially when you already have a live Spoke Phone setup***, we suggest using Postman Desktop, to help you simulate the Spoke Phone events and make testing the integrations faster:

- Postman Desktop.
    - Whilst using two or three phones with different numbers is the best way to test your power automate flows, we understand sometimes this is not practical, or you already have a Spoke Phone production system live and being used (which would trigger updates from live users to Dynamics).
    - Postman allows you to simulate Spoke Phone events, without having to run the Spoke Phone App and register the Spoke Phone webhooks to Power Automate flows.
    - Installing the Desktop version of this application makes things easier with support for dynamic variables.
    - Spoke Phone has provided a Postman Collection which includes a set of sample requests to simulate the call events required for testing the integration.

Power Automate Dynamics365 Templates Overview

User Journeys

Overview

The integrations rely on Webhooks for fast notification and processing of new data resulting from interactions with either Dynamics365 or Spoke Phone. The data is usually processed within seconds of the event occurring. The Power Automate Spoke Phone to Dynamics365 example flows are designed to:

- Synchronize contact data from the CRM into the Spoke phonebook so that Spoke Phone users have the most up to date contact information available from the CRM.
- Update the CRM with new phone numbers for new contacts as well as new phone numbers assigned to existing contacts.
- Update the CRM with phone call data such as call records, voice notes, mobile form data, etc.
- The above mentioned are the most common scenarios the example flows cover, we encourage customers to alter them and extend them to fit the unique needs of the business.

The flows are structured as follows:

SpokePhone: Authenticate Spoke API
This 'Child Flow' is used to authenticate to Spoke Phone so other flows can post to the Spoke Phone API.

SpokePhone: Manual - Dynamics365 contacts to Spoke Phonebook
A manually triggered flow to transfer all Dynamics contacts to Spoke Phonebook.

SpokePhone: Dynamics365Webhook - Delete contact to Spoke Phonebook
Dynamics webhook event on a contact delete, synchronizing to Spoke Phonebook.

SpokePhone: Dynamics365Webhook - Create or update contact to Spoke Phonebook
Dynamics webhook event on a contact update OR add, synchronizing to Spoke Phonebook.

SpokePhone: ChildFlow - Log phone call to Dynamics365
A child flow, used to log a phone call record. Used by the below flows to ensure a call log exists.

SpokePhone: SpokeWebhook - Call Ended to Dynamics365
Spoke Phone webhook event when a phone call ends, logs the phone call record if it doesn't already exist AND if the assigned contact exists in Dynamics.

SpokePhone: SpokeWebhook - Contact assigned to Dynamics365
Spoke Phone webhook event when a new contact is assigned to an unknown number. Adds a new contact, or updates an existing contact, and then logs a phone call record to the new or updated contact.

SpokePhone: SpokeWebhook - Call note added to Dynamics365
Spoke Phone webhook event when a new voice note is added to a call. The textual voice note is added against the contact timeline and against the call log timeline as well as updating the call log description.

SpokePhone: SpokeWebhook - Call highlight recording to Dynamics365
Spoke Phone webhook event when a call highlight recording audio file is available. The audio file is added to the call log timeline against a new note.

SpokePhone: SpokeWebhook - Form Submitted to Dynamics365
Spoke Phone webhook event when a form is submitted, logs the form data to the phone call record. Will create a new phone call record if one doesn't exist AND the assigned contact exists in the CRM.

SpokePhone: SpokeWebhook - Form Submitted to Excel OneDrive
Spoke Phone webhook event when a form is submitted, logs the form data to an excel spreadsheet. The process flattens all form data into an individual row and handles new form fields. The excel file is a no code solution for reporting and extensive analysis.

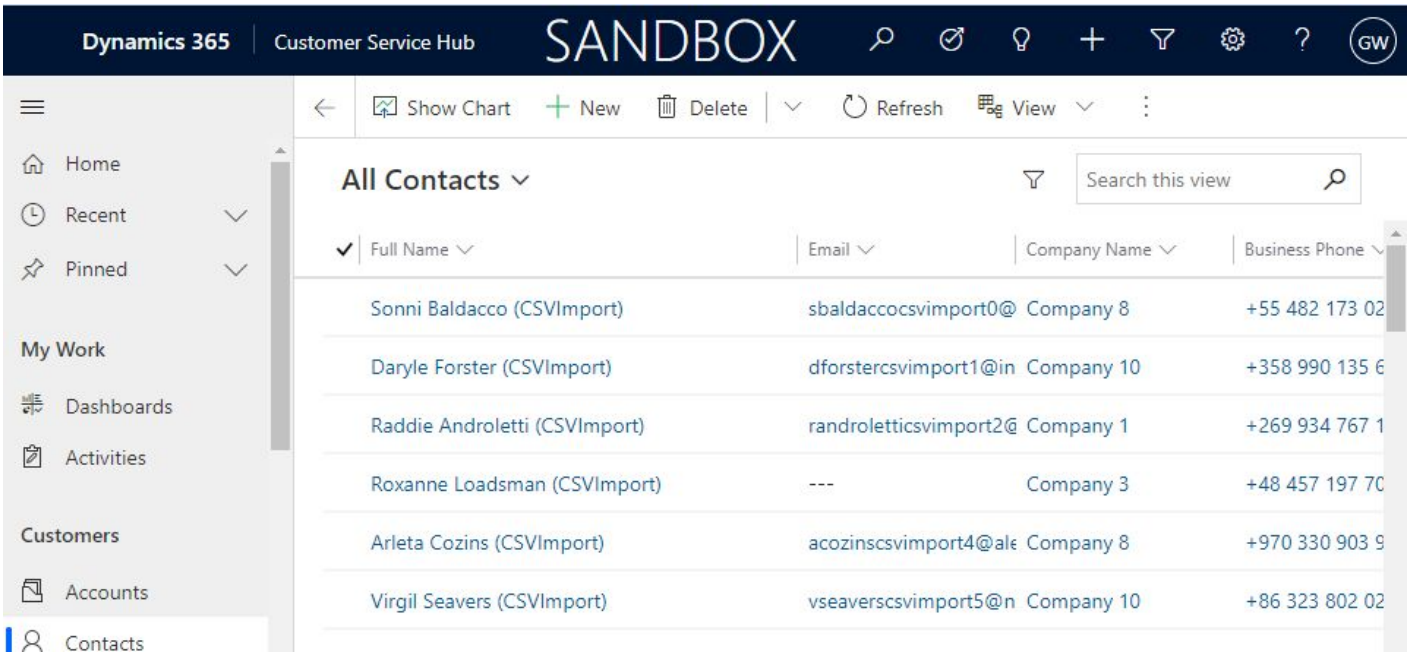
Using the Power Automate flows

Installing the Power Automate flows.

Environment Considerations

The Microsoft Power Automate boilerplate flows connect to the ‘current’ Microsoft cloud environment. If you wish to test the flows in the ‘Sandbox’ environment, ensure you import the solution into sandbox, which will in turn integrate into the CRM within that sandbox environment.

Ensure your ‘Sandbox’ CRM installation is the same as your production environment in terms of customizations, to correctly test for real-world scenarios.



Please See “Suggested Dynamics customizations” to get full value from the flows

Importing the Flows

1. Navigate to Power Automate and ensure the appropriate environment is selected.
2. Navigate to ‘solutions’ → ‘Import’ and select the “SpokePhoneDynamics365” zip file
3. Remain on the page until the solution has been imported.
4. Navigate to the solution ‘SpokePhone: Dynamics365’.

Testing the Power Automate Dynamics365 contact integrations

Set up the Spoke Phone API

- Login to the Spoke Phone Account portal (https://account.spokephone.com/)
- Navigate to Other → Developers
- Add a new API key and name it, e.g. “Power Automate Dynamics Integration”.
- The clientID and client secret are your API credentials used in the power automate solution.

Complete the Power Automate flow setup for Dynamics contact integrations

The Microsoft Power Automate Flows will need to be opened, to set your environment variables or instantiate the connections to MS CDS services.

Please follow these steps:

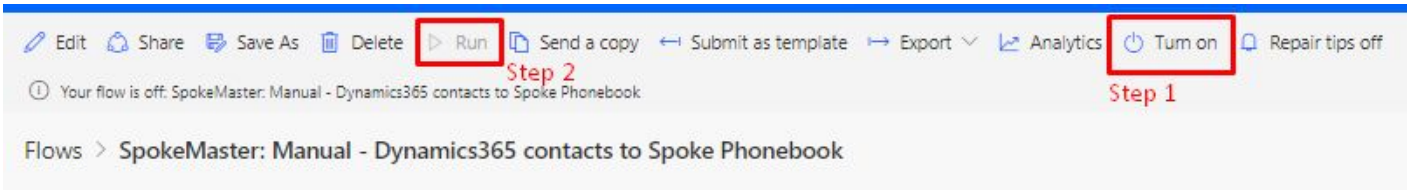
SpokePhone: Authenticate Spoke API ● Edit this flow and edit the authentication variables for your Spoke installation: ○ Spoke-ClientID, Spoke-Secret ● Ensure you save and close to complete the setting of your authentication parameters.	❑ **Set API Key** ❑ Save flow
SpokePhone: Manual - Dynamics365 contacts to Spoke Phonebook ● Edit this flow, when first opening you may need to select the connection to MS CDS. ● Clicking save will complete the connection process, allowing the flow to function.	❑ Save flow
SpokePhone: Dynamics365Webhook - Create or update contact to Spoke Phonebook ● Edit this flow, when first opening you may need to select the connection to MS CDS. ● Clicking save will complete the connection process, allowing the flow to function.	❑ Save flow
SpokePhone: Dynamics365Webhook - Delete contact to Spoke Phonebook ● Edit this flow, when first opening you may need to select the connection to MS CDS. ● Clicking save will complete the connection process, allowing the flow to function.	❑ Save flow

Note: There is no need to edit any other flows at this stage.

Build and load the Spoke phonebook

Firstly, we transfer all the CRM contacts into the Spoke phonebook.

1. Open the flow: “SpokePhone: Manual - Dynamics365 contacts to Spoke Phonebook”
2. Click ‘Turn on’ and the ‘Run’ from the toolbar:

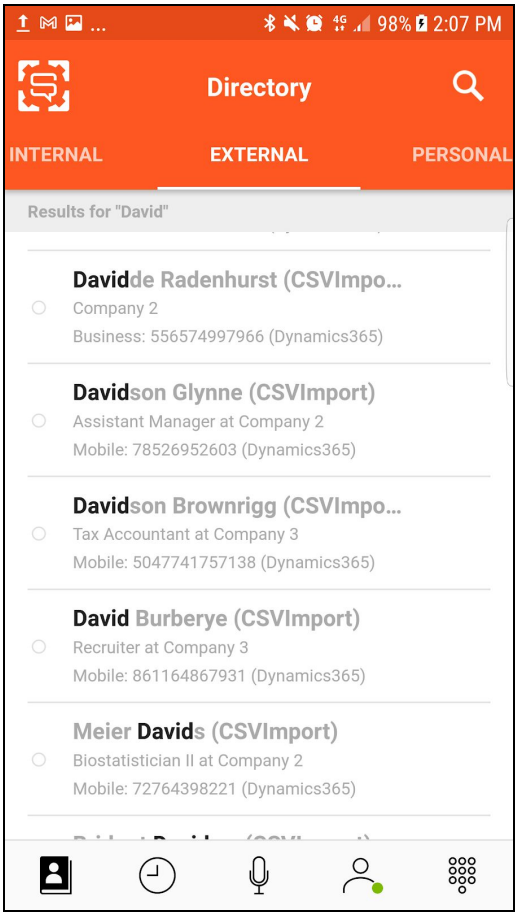


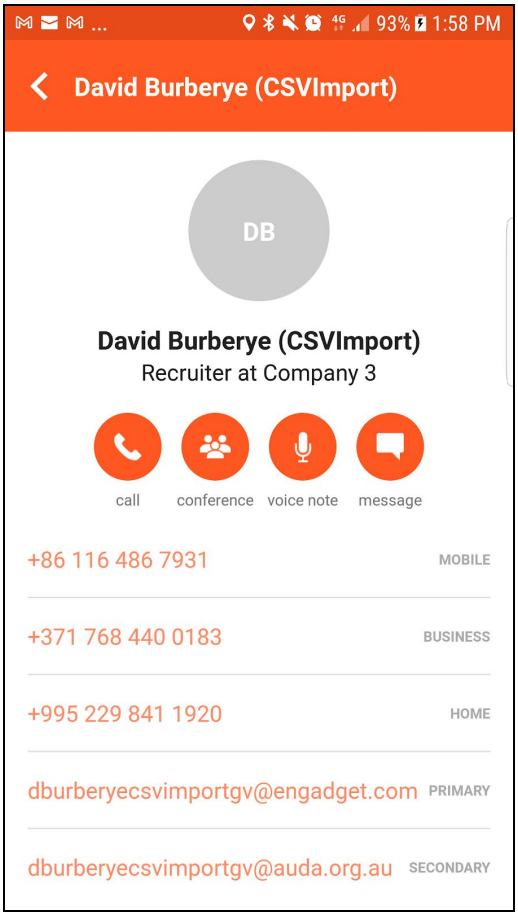
3. Leave the option ‘deletephonebookwithnorebuild’ switched off and click ‘Run flow’
4. Observe the flow history.

   After a successful run, the flow has taken all ‘active’ contacts from your CRM and loaded them into the Spoke phone book named ‘Dynamics365’.

   The system takes roughly 15-20 minutes per 10,000 contacts but varies depending on HTTP availability of the CRM and other system environment factors.

5. Test that the contacts are loaded into the Spoke Phonebook by navigating the Spoke mobile App to: ‘Directory’ → ‘External’ and typing in a search term three characters or longer.





Test the CRM contact webhooks

Now we need to ensure that when the CRM contacts change, the appropriate message is sent to update the Spoke phonebook accordingly. Two Power Automate flows handle these updates:

"SpokePhone: Dynamics365Webhook - Create or update contact to Spoke Phonebook"
"SpokePhone: Dynamics365Webhook - Delete contact to Spoke Phonebook"

Confirm setup

- Confirm you have the correct API keys setup in the Microsoft Power Automate flow: ***"SpokePhone: Authenticate Spoke API"***.
- Confirm you are in the expected environment (sandbox vs production).

Test 1: Add new contact to Dynamics 365

User Journey	Add new contact in Dynamics 365 - copy contact to Spoke Phonebook
Run the test	Add a new 'test' contact in the CRM, populate all of the fields that Spoke Phone supports: Firstname, Lastname, Job Title, Account Name (contact or account), Mobile, Business, Home, Telephone_3, email, email_2, email_3.
Validate	Observe the new contact in the Spoke Phone app by navigating to Directory → External and searching for the contact you just added.
Notes	You can observe the Power Automate run log if you wish to see the data that was processed during this test.

Test 2: Edit a contact in Dynamics 365

User Journey	Edit contact in Dynamics 365 - update contact in Spoke Phonebook
Run the test	Change any of the Spoke Phone supported contact fields in the CRM and save.
Validate	Observe the updated data on the contact in the Spoke Phone app.
Notes	You will need to enter a new search term e.g. 'abc' to 'reset' the previous search results and data. Perform a fresh search for your test contact, to 'refresh' the data.

Test 3: Add new contact to CRM

User Journey	Deactivate a contact in Dynamics 365 - remove contact from Spoke Phonebook
Run the test	1. Deactivate a contact in the CRM and save. 2. Reactivate the contact in the CRM and save.
Validate	1. Observe that the contact can no longer be found in the Spoke Phone app. 2. Observe the contact repopulating.

Notes	

Test 4: Add new contact to CRM

User Journey	Delete a contact in Dynamics 365 - remove contact from Spoke Phonebook
Run the test	Delete a contact in the CRM and save.
Validate	Observe that the contact can no longer be found in the Spoke Phone app.
Notes	

Test 5: Merge two contacts in the CRM

User Journey	Delete a contact in Dynamics 365 - remove contact from Spoke Phonebook
Run the test	Set up two new test contacts and merge them in the CRM.
Validate	Observe that the Spoke phonebook results in only the final merged contact being available.
Notes	If you have a duplicate contact in the CRM, the flows will also handle the ‘merge’ feature appropriately. This can happen if users add new contact numbers to the CRM as new contacts as opposed to selecting to update an existing contact.

Suggested Dynamics Customizations

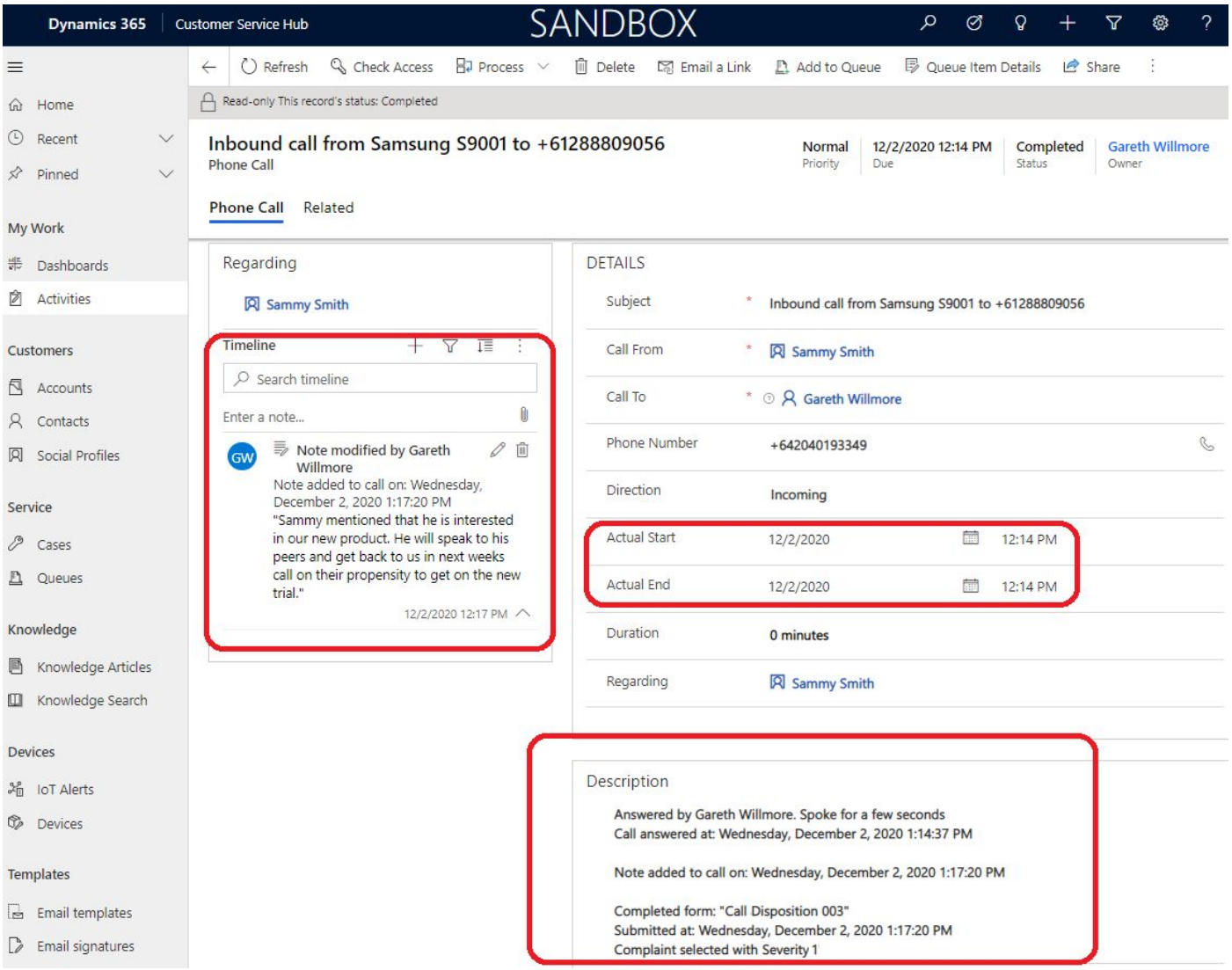
The following recommendations help to get the most out of Spoke Phone data sent to the CRM. Please make these customizations as you see fit in respect to your environment configuration. These are intended as hints as to where to make the changes and not detailed instructions for Dynamics customization.

Phone Call Entity:

Field_Actual Start: Customize this field to have the 'format' of 'Date and Time'.
Field_Actual End: Customize this field to have the 'format' of 'Date and Time'.
Form_Phone Call:

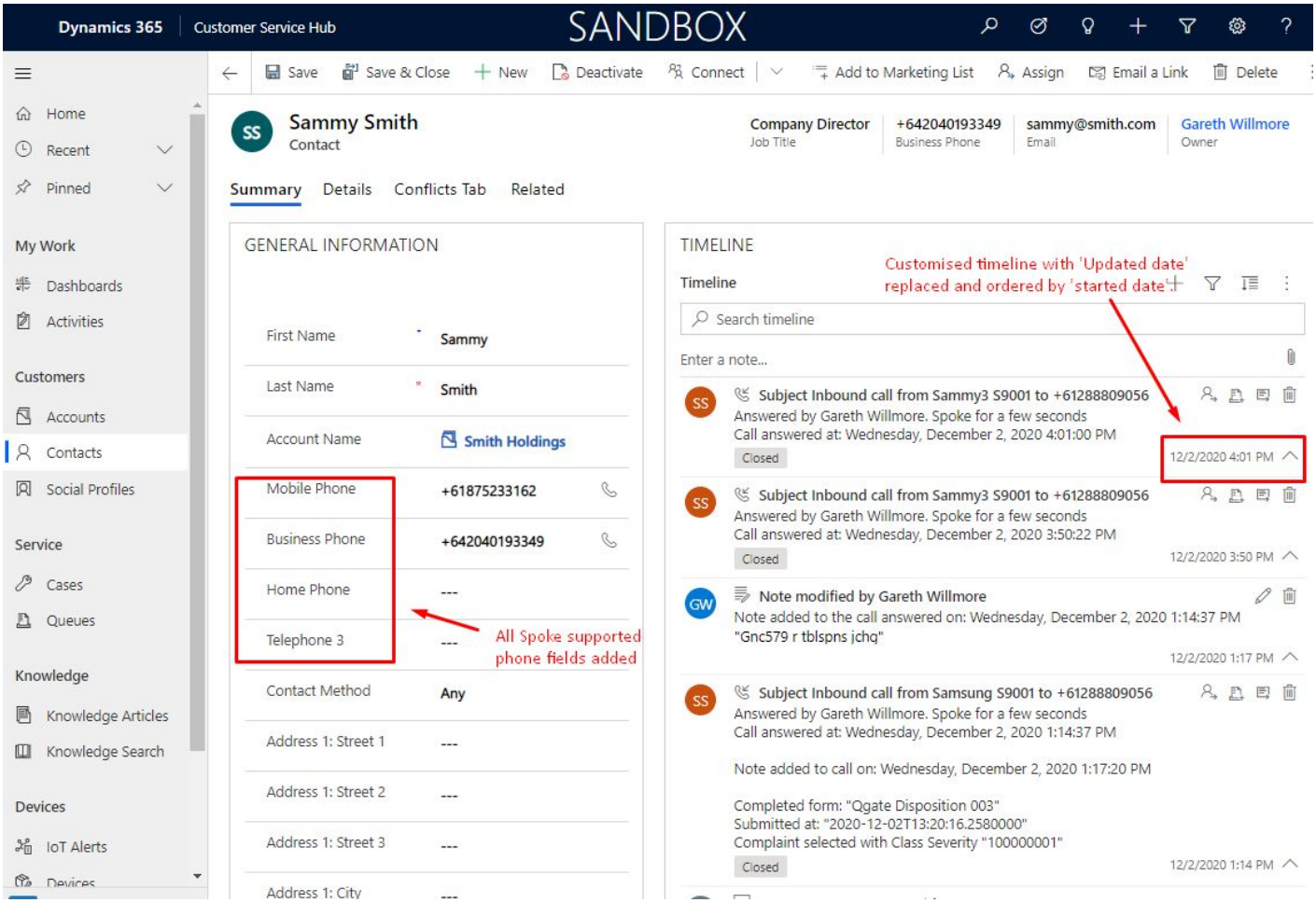
- Insert a 'timeline' form element to show the 'notes' attached to the phone call. E.g. voice notes, form response data and recorded call highlight audio.
- Edit the 'description' field and check the box 'Automatically expand to use available space'
- Add the fields 'Actual Start' and 'Actual End'

After these customizations a phone call will look similar to this:



Contact Entity Form:

- By default, the contact form contains a timeline sorted by the 'last updated' field. This will allow users to see those phone calls that have NOTES or FORMS submitted with them.
- We suggest editing the timeline properties and selecting '**Expand all records by default**'.
- A phone call record is updated if a form is submitted later, or a voice note is added later.
- Edit the timeline on the form and choose '**sort activities by' → "Start Date**", if you wish the phone call activities to be organized by when the phone call started, as opposed to last updated.
- **Note**: The contact form by default will be 'default form for the interactive experience', however, your Dynamics365 setup may be quite different.
- **Note**: this change will affect all activities against a contact.
- It is also possible to edit the contact form timeline to show the 'date started' rather than the 'date last updated'. This requires the addition of a form against each activities you need to customize:
    - To do this, you can add a new form of type 'card' to the 'phone call' entity (table in new terminology). You can add the data you would like to show on the timeline in this 'card'. The first column of the 'header' is the timeline subject, the second column is the date at the bottom right of the timeline activity - this is where you can add 'Start Date'. You will then also want 'description' in the main body of the new card form.
    - Ensure the 'form order' for the 'Card form' set is edited so that your new form is at the top.
    - You will then need to edit the 'contact form' → 'timeline' properties and select the new form you created in the 'Phone Call' activity type.



Testing the Power Automate Dynamics365 Spoke Phone event integrations

Complete the Power Automate flow setup for Spoke Webhooks

The Microsoft Power Automate Flows will need to be opened, to set your environment variables or instantiate the connections to MS CDS services.

Please follow these steps:

SpokePhone: ChildFlow - Log phone call to Dynamics365 • Edit this flow, when first opening you may need to select the connection to MS CDS. • Open up “Initialize and Set timeZone” and set the value to be your preferred timezone. • In the node ‘dynamicsUserId’ you will need to enter the CRMs ID of the ‘default user’ you would like to associate with Spoke Phone. When the Spoke Phone user cannot be matched (by email address) to a dynamics user, this default user will be assigned to the phone call. • Clicking save will complete the connection process, allowing the flow to function.	❑ **Set timezone** ❑ **Set default dynamics userId** ❑ Save flow
SpokePhone: SpokeWebhook - Call Ended to Dynamics365 • Edit this flow, when first opening you may need to select the connection to MS CDS. • When saving this flow the ‘HTTP Request’ post URL is generated. This URL will be used to register the webhook in the Spoke Phone account portal.	❑ Save flow
SpokePhone: SpokeWebhook - Contact assigned to Dynamics365 • Edit this flow, when first opening you may need to select the connection to MS CDS. • When saving this flow the ‘HTTP Request’ post URL is generated. This URL will be used to register the webhook in the Spoke Phone account portal.	❑ Save flow
SpokePhone: SpokeWebhook - Form Submitted to Dynamics365 • Edit this flow, when first opening you may need to select the connection to MS CDS. • Open up “Initialize and Set timeZone” and set the value to be your preferred timezone. • Optional: If you have mobile forms, you can edit this area to perform any custom processing for specific form references. • When saving this flow the ‘HTTP Request’ post URL is generated. This URL will be used to register the webhook in the Spoke Phone account portal.	❑ **Set timezone** ❑ Optional: Mobile form processing ❑ Save flow
SpokePhone: SpokeWebhook - Call note added to Dynamics365 • Edit this flow, when first opening you may need to select the connection to MS CDS. • Open up “Initialize and Set timeZone” and set the value to be your preferred timezone. • When saving this flow the ‘HTTP Request’ post URL is generated. This URL will be used to register the webhook in the Spoke Phone account portal.	❑ **Set timezone** ❑ Save flow
OPTIONAL - ***SpokePhone: SpokeWebhook - Call highlight recording to Dynamics365*** ***NOTE: If you have a policy not to store call audio in your CRM, then you can simply disable this flow and not set up the webhook in the Spoke Phone account portal.*** • Edit this flow, when first opening you may need to select the connection to MS CDS. • Open up “Initialize and Set timeZone” and set the value to be your preferred timezone.	[optional]

• When saving this flow the 'HTTP Request' post URL is generated. This URL will be used to register the webhook in the Spoke Phone account portal.	❑ Set timezone ❑ Save flow

Set up the Spoke Phone Webhooks

- Login to the Spoke Phone Account portal (https://account.spokephone.com/)
- Navigate to Other → Developers → Webhooks
- Add a new Webhook:
    - Retrieve the URL from Power Automate:
        - In the Power Automate solution, edit the flow ***"SpokePhone: SpokeWebhook - Contact assigned to Dynamics365"***.
        - Copy the URL from the first node in the flow:

        - You may close the flow without saving.
    - Enter a description, e.g. "Power Automate: New/updated contact to Dynamics 365".
    - Add two event subscriptions:
      "Call.contact_assigned_with_add"
      "Call.contact_assigned_with_update"
    - Click save.
- Add a new Webhook:
    - Retrieve the URL from power automate:
        - In the Power Automate solution, edit the flow "***SpokePhone: SpokeWebhook - Call Ended to Dynamics365***".
        - Copy the URL from the first node in the flow. You may close the flow without saving.
    - Enter a description, e.g. "Power Automate: Call log to Dynamics 365".
    - Add one event subscription:
      "Call.ended"
    - Click save.
- Add a new Webhook:
    - Retrieve the URL from power automate:
        - In the Power Automate solution, edit the flow "***SpokePhone: SpokeWebhook - Call note added to Dynamics365***".
        - Copy the URL from the first node in the flow. You may close the flow without saving.
    - Enter a description, e.g. "Power Automate: Note added log to Dynamics 365".
    - Add one event subscription:
      "Call.note.created"
    - Click save.
- Add a new Webhook:
    - Retrieve the URL from power automate:
        - In the Power Automate solution, edit the flow "***SpokePhone: SpokeWebhook - Form Submitted to Dynamics365***".
        - Copy the URL from the first node in the flow. You may close the flow without saving.
    - Enter a description, e.g. "Power Automate: Form submitted to Dynamics 365".
    - Add one event subscription:
      "Call.form.submitted"
    - Click save.

- ***OPTIONAL***: Add a new Webhook:
    - Retrieve the URL from power automate:
        - In the Power Automate solution, edit the flow "***SpokePhone: SpokeWebhook - Call Highlight Recording to Dynamics365***".
        - Copy the URL from the first node in the flow. You may close the flow without saving.
    - Enter a description, e.g. "Power Automate: Call highlight recording to Dynamics 365".
    - Add one event subscription:
      "Call.form.submitted"
    - Click save.

Set up a Spoke Phone test form

To test the form submitted process and log into the CRM call log that a form has been submitted, we can install a new test form into Spoke Phone.

- Login to the Spoke Phone Account portal (https://account.spokephone.com/)
- Navigate to Other → Developers → Form Definitions
- Add a new mobile form:
    - Name the form: E.g. "Call disposition"
    - Enter a description.
    - Enter a form reference: E.g. "Disposition001"
    - Select 'Conversation' type for the form.
    - Toggle the 'Make this form active' option to ON.
    - Enter the Form Data Definition:
        - Open the 'Call Disposition Sample Data.txt' file from the Spoke Phone pack.
        - Copy and paste the contents.
    - Enter the Form Display Definition:
        - Open the 'Call Disposition Sample Display.txt' file from the Spoke Phone pack.
        - Copy and paste the contents.
    - Click save.
    - The new form will now be available during calls and on the call history in the Spoke Phone App.
    - We will test this form in 'Test 4' below.

Optional: Set up an excel log

You may optionally wish to set up an excel workbook in oneDrive that generically logs form responses. Spoke Phone have created an example flow showing how this can be done with Power Automate, however we encourage you to fully understand the impact of using this, as it uses loops to process the submitted form. Note, it is very easy to disable this flow if it causes any issues.

When a form is created with many elements nested in objects and arrays, this process can consume many 'Billable Actions' as a result of needing to loop through to expand the node set. Extra billable actions can be purchased, however overage can result in flows becoming progressively throttled or even disabled by Microsoft. At the time of writing this won't incur charges, but may effect your other flows such as call logging.

Setup the template file

To use this flow you will need to have the template excel file in Onedrive. The file can be found here:

https://sp-static-assets-prod.s3.amazonaws.com/sampleIntegrations/microsoft/powerAutomate/SpokePhoneFormsTemplate.xlsx

Configure the Power Automate flow

OPTIONAL - ***SpokePhone: SpokeWebhook - Form Submitted to Excel OneDrive***
 	
NOTE: This is an optional example flow showing how a form can be dynamically processed, flattened and saved in Excel. There are extra steps to see this working:
- Setup1: Ensure you have the template file "SpokePhoneFormsTemplate.xlsx"
- NOTE: Confirm you understand the billable 'Actions' within the flow and the consequence of exceeding limits. Since this flow loops over arrays and objects, a form can create a large number of billable actions if it contains many nested objects and arrays. If you have large forms containing many objects and arrays, which are likely to generate many form submissions, we suggest building a JSON to flat array parser in Power Apps.
- NOTE: Data elements nested in objects and arrays are renamed in the flattened structure. The names are a compound of the parent objects and array iterations. Object parents separated by "/" and array parents by "|" and the array iteration, e.g.
  - `"a":{ "a1":"value a1"}` translates to column name "a/a1"
  - `"a":[{"b"},{"b","c"}]` translates to columns "a|0|b" "a|1|b" "a|1|c"
  Note, 3 fields are found since b is found in two different array positions and the data is considered a different column.
- The column names are not logically ordered. They are ordered by the 'nested level' since the flow processes all fields in a nested level and expands the field list, before processing the next nested level.
- The flow is also designed to handle new columns if a form with the same reference is submitted. Any newly found columns are appended to the end of the table.
 	
● Edit this flow, when first opening you may need to select the connection to MS CDS.
● Open up "***Initialize and Set filePath***" and set the value to the path you wish to store form responses file in.
● Open up "***Initialize and Set fileName***" and set the value to the name of the file you wish to store form responses in.
	[optional]
	
	❑ Set filepath
	❑ Set filename
	❑ Set timezone

• Open up "***Initialize and Set timeZone***" and set the value to be your preferred timezone.
• Open up "***setup File → Create file if not exist → Get file content using path***" and ensure this points to the Spoke Phone template excel file mentioned above.
• Open the Excel flow nodes in order to set your environment connection. Open up each node listed below and edit ***"Location"*** and ***"Document Library"*** values:
  ○ "*Create new worksheet or update existing worksheet with new columns*"
  ■ → "***Get Worksheets***"
  ■ → "*Create worksheet or update existing worksheet*"
  • → "*If Yes*"
  ○ → "***Create worksheet***"
  ○ → "***Create table***"
  ○ → "***Log failed form reference with special characters***"
  ○ → "*Apply to each*"
  ■ → "***Add a key column to a table***"
  ○ → "***Add a key column to a table***"
  • → "*If No*"
  ○ → "*Check for new columns*"
  ■ → "***Add a key column to a table 2***"
  ○ "*Add extracted form data into a new row in the forms reference worksheet*"
  ■ → "***Add a row into a table 2***"
• When saving this flow the 'HTTP Request' post URL is generated. This URL will be used to register the webhook in the Spoke Phone account portal.
	❑ Setup template
	
	❑ Edit Excel node 1
	
	❑ Edit Excel node 2
	❑ Edit Excel node 3
	❑ Edit Excel node 4
	
	❑ Edit Excel node 5
	❑ Edit Excel node 6
	
	❑ Edit Excel node 7
	
	❑ Edit Excel node 8
	
	❑ Save flow

Set up the Spoke Phone Webhook for forms to Excel

- Login to the Spoke Phone Account portal (https://account.spokephone.com/)
- Navigate to Other → Developers → Webhooks
- ***OPTIONAL***: Add a new Webhook:
  - Retrieve the URL from power automate:
    - In the Power Automate solution, edit the flow "***SpokePhone: SpokeWebhook - Form submitted to Excel OneDrive***".
    - Copy the URL from the first node in the flow. You may close the flow without saving.
  - Enter a description, e.g. "Power Automate: Form submitted to Excel".
  - Add one event subscription:
    "Call.form.submitted"
  - Click save.

NOTE: Keeping this webhook separate from the form submitted to Dynamics, allows you to manage the flows independently and turn them off by disabling the webhooks should you wish too.

Choosing the test method

Testing the Spoke Phone integrations with the Spoke Phone App

This method is recommended if you are new to Spoke Phone and have no live users (or very few).

Alternative Test Method: Testing the Spoke webhook integrations with Postman

If you already have a Spokephone live, or if you do not have access to multiple devices, you may want to test the integration without hooking it up to Spoke Phone Webhooks so that your tests are not buried in the noise of live events. You can run a large majority of tests using Postman to simulate Spoke Phone webhooks. A section in this article guides you through using Postman to simulate webhook events.

Testing the Spoke Phone integrations with the Spoke Phone App

NOTE: If you already have a Spokephone live with an active user base, you <u>may</u> want to test the integration without hooking it up to Spoke Phone. You can run a large majority of tests using Postman to simulate Spoke Phone webhook events. If this applies to you please navigate to the next section: "Testing the Power Automate flows with Postman"

Preparation and device requirements

To test the Spoke Phone webhooks you will need to make phone calls to a user in Spoke Phone from another device. Ideally, you can make calls from at least one other 'unknown' number that is not in the Spoke phonebook.

If your CRM contains the numbers you wish to test from, you may wish to remove those numbers and re-add them after the tests have been performed.

Running the tests with the Spoke Phone App: Recommended test script

Mobile1: A mobile phone with the Spoke Phone App installed and an administrator signed in.
Mobile2: A mobile phone with a number that is not currently stored in the CRM.

Test 1: Add new contact to CRM

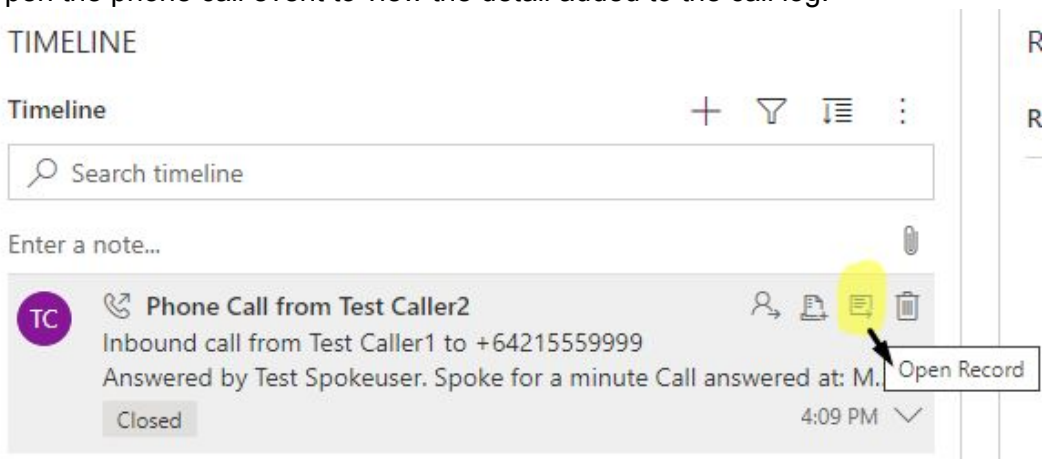
User Journey	When a Spoke Phone user receives a call from a number that does not exist in the Spoke Phonebook (and therefore the CRM), the Spoke Phone user has the option to 'Add to CRM' or 'Select an existing contact to associate the number with'. This test runs the 'add new contact to CRM' option.
Flow used	***SpokePhone: SpokeWebhook - Contact assigned to Dynamics365***
Run the test	• Open the Spoke Phone app on Mobile1 • Make a call to the Spoke Phone number from Mobile2 • Observe the number calling *Note: If this shows a contact name, you should delete the number from that contact in the CRM, and try again.* • End the call from either phone. • On Mobile1, click the 'call history' icon at the bottom of the Spoke Phone app. • Select the phone call record just made. • Click 'Add to CRM'. • Type at least 3 characters to search for an existing contact and click 'Next' • We are simulating a new contact, so click 'Add a new lead/contact' • Enter firstname:"Test", lastname:"Caller1" and click 'Save'

Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the ‘run history’, which should show the run you just triggered as ‘Succeeded’. You can open that run history and observe the flow state, variables and data as it ran through the steps. In your Dynamics Sandbox Environment (that you connected the flows to when importing the solution and saving the flows) you should now be able to open up the ‘contacts’ in Dynamics CRM and observe the newly added contact ‘Test Caller1’. Open up that contact and observe the detail added to the contact. Also observe a ‘phone call’ record added to that contact. Open up the phone call record and observe the detail added.
Notes	NOTE: your dynamics ‘contact’ and ‘phone call’ forms can be customized and may not show all fields supported in the Power Automate flows.

Test 2: Add unknown number to existing contact in CRM

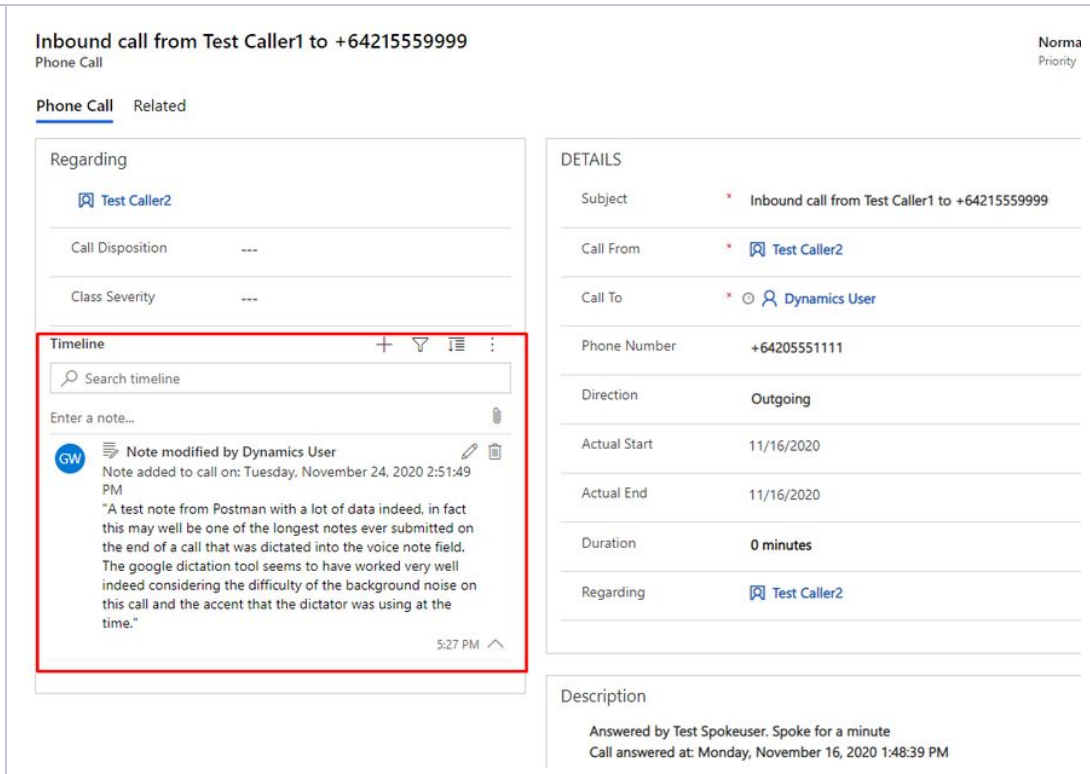
User Journey	When a Spoke Phone user receives a call from a number that does not exist in the Spoke Phonebook (and therefore the CRM), the Spoke Phone user has the option to 'Add to CRM' or 'Select an existing contact to associate the number with'. This test runs the 'add new number to existing contact in CRM' option.
Flow used	***SpokePhone: SpokeWebhook - Contact assigned to Dynamics365***
Run the test	Pre-setup: We need a number that is unknown in the CRM. ● Edit the mobile phone number in the CRM against the "Test Caller1" contact just created. Change the number to any other number. 1. Open the Spoke Phone app on Mobile1 2. Make a call to the Spoke Phone number from Mobile2 3. Observe the number calling *Note: If this shows a contact name, you should delete the number from that contact in the CRM, and try again.* 4. End the call from either phone. 5. On Mobile1, click the 'call history' icon at the bottom of the Spoke Phone app. 6. Select the phone call record just made. 7. Click 'Add to CRM'. 8. Type: "Caller1" to search for the contact and click next. 9. Select the contact 'Test Caller1' to associate the unknown number to the contact.
Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the 'run history', which should show the run you just triggered as 'Succeeded'. You can open that run history and observe the flow state, variables and data as it ran through the steps. Note that this time, under the "Add contact to CRM or update…" node (at the bottom of the flow) the flow went down 'Case 2' in the flow control, running the update routine, as opposed to the add new contact routine. In your Dynamics Sandbox Environment open up the 'contacts' in Dynamics CRM and open the contact 'Test Caller1'. You should now see the new number added to the 'Business Phone' Field. Rerun the test to show the 'new number' getting added to the other phone fields. See the notes below for more detail.
Notes	The new unknown number has been added to the contact and Spoke Phone will now recognize it (the contact integration updates the Spoke phonebook with the new data). You can rerun this test to show when a new number is added to an existing contact, it adds that new number to the first empty field it finds in this order: mobile, business, home, telephone3. Note that if telephone3 contains a number already, the flow is set up to update it with the new number, since the new number is most likely more useful. If you have more than 1 mobile phone, you can test calling from other unknown numbers and adding them to the test contact.

Test 3: Add a call log for a known caller

User Journey	When a call is received from a number that is found in the Spoke phonebook, or a call is made to a contact in the Spoke phonebook, when that call ends the information is written to contact that was matched. The call is logged against the CRM contact and the CRM 'user' that matches the Spoke Phone users email address that made or received the call. If the email is not found, the default CRM user is used (as configured in the flow installation guide)
Flow used	***SpokePhone: SpokeWebhook - Call Ended to Dynamics365***
Run the test	1. Open the Spoke Phone app on Mobile1 2. Make a call to the Spoke Phone number from Mobile2 3. Observe the number calling and observe it has matched contact "Test Caller1" 4. End the call from either phone.
Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the 'run history', which should show the run you just triggered as 'Succeeded'. Note that both the 'contact assigned' and the 'call ended' flows call upon the child flow to log the phone call record. You can open the child flow and observe the successful runs of the phone call log. In your Dynamics Sandbox Environment open up the 'contacts' in Dynamics CRM and open the contact 'Test Caller1'. You should see the 'phone call' added to the timeline. Click on this to observe more text in the description field. Open the phone call event to view the detail added to the call log: 
Notes	The date in the timeline shows when that 'phone call' event was last edited, not the actual date of the call. The date of the call is stored in the call description, as well as in the 'due date' and the 'actual start' and 'actual end' datetime fields. You can run this test as many times as you like, each run will add a new call log.

Test 4: Add a note to a call

User Journey	In the Spoke Phone app, the user can add a ‘voice note’ to a call. This is done post call from the call history screen. When a voice note is added, the webhook event ‘Call note created’ is triggered. Many voice notes can be added to the same call. The note is added to the phone call event in the CRM.
Flow used	***SpokePhone: SpokeWebhook - Call note added to Dynamics365***
Run the test	
Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the ‘run history’, which should show the run you just triggered as ‘Succeeded’. In your Dynamics Sandbox Environment open up the ‘contacts’ in Dynamics CRM and open the contact ‘Test Caller2’. On the contact timeline you can observe the note added as a new ‘note’ activity. The full voice note that was dictated (or typed) is populated on the call record. Click on the ‘phone call’ in the timeline to expand the detail, notice that the phone call may have moved up in the timeline, since the record has been updated, and the activities are ordered by ‘last updated’ date time. In the phone call detail, you can observe the description is expanded with a comment “Note added to call on: {{datetime}}”. This makes reference that this call had a note added to it. Fully open up the call record by clicking the ‘open record’ icon (as noted in test 3). Observe the details of the call record.
Notes	By Dynamics default this screen does not show notes added to a phone call. You can customize this form in Dynamics admin to show notes added to phone calls. This is a cleaner way to store notes and is a suggested practice. The customization can be seen below where a ‘timeline’ object is added to the phone call form. The flow adds this by default and when you make the customization it will show up here, at which point you may choose to remove the part of the flow that adds the note to the contacts timeline.


Inbound call from Test Caller1 to +64215559999
Phone Call
Norma
Priority
Phone Call
Related
Regarding
Test Caller2
Call Disposition

Class Severity

Timeline
Search timeline
Enter a note...
GW
Note modified by Dynamics User
Note added to call on: Tuesday, November 24, 2020 2:51:49 PM
"A test note from Postman with a lot of data indeed, in fact this may well be one of the longest notes ever submitted on the end of a call that was dictated into the voice note field. The google dictation tool seems to have worked very well indeed considering the difficulty of the background noise on this call and the accent that the dictator was using at the time."
5:27 PM
DETAILS
Subject
Inbound call from Test Caller1 to +64215559999
Call From
Test Caller2
Call To
Dynamics User
Phone Number
+64205551111
Direction
Outgoing
Actual Start
11/16/2020
Actual End
11/16/2020
Duration
0 minutes
Regarding
Test Caller2
Description
Answered by Test Spokeuser. Spoke for a minute
Call answered at: Monday, November 16, 2020 1:48:39 PM

Test 5: Form submitted to a call

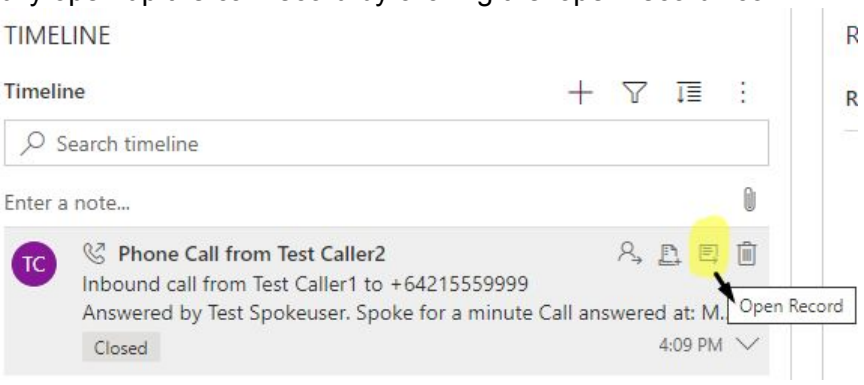
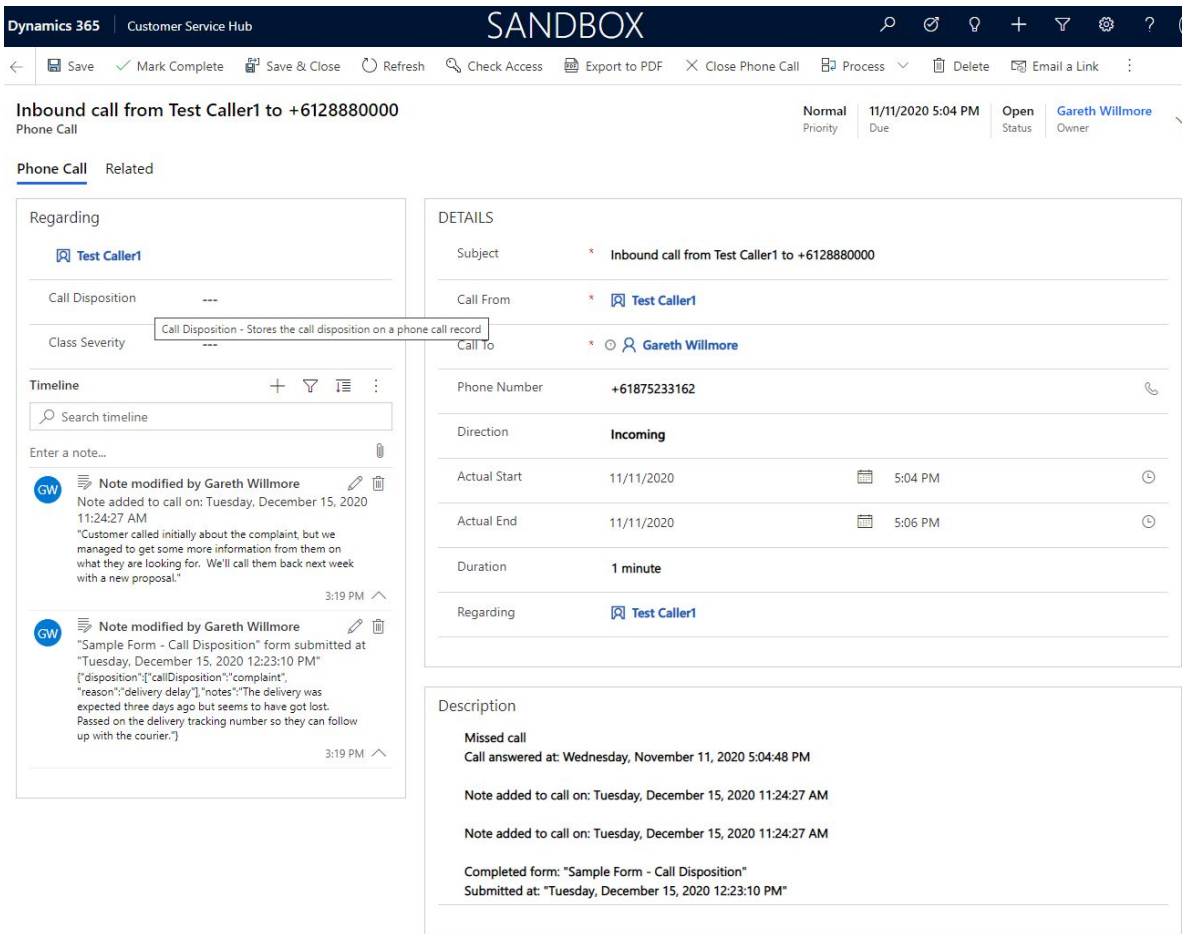
In the Spoke Phone app, the user can submit a mobile form against a call history record. This can be started within the call to begin filling out details whilst talking to the caller, but is submitted post-call from the call history screen. When a form is submitted, the webhook event 'Call form submitted' is triggered.

The flow contains a 'generic' process, which results in two things: the phone call record description is updated with information that a form was completed, and a note being added to the phone call record 'timeline' in the CRM with detail from the submitted form. ***NOTE***: To see this 'note' on the 'phone call record' the default Dynamics365 phone call screen should be updated to include the 'timeline' element in the 'phone call' entity's forms.

The flow first checks for the 'FormReference' before running the processing of the form. Since mobile forms are customized by you, the data needs to be processed in a custom way if any logic is required or if particular fields need to be processed.

Mobile form processes can be built to save or update CRM data based on the form responses as needed. The Power Automate form processing flow may also be edited to post that mobile form data to other systems such as Excel spreadsheets, trigger an email, create a CRM contact case or issue, etc.

Step	
Flow used	***SpokePhone: SpokeWebhook - Form Submitted to Dynamics365***
Run the test	Follow the steps in section Setup a Spoke Phone test form if you haven't already 1. On Mobile1, click the 'call history' icon at the bottom of the Spoke Phone app. Note: You can make a test call and simply hang up to get a new call history item. You will only be able to submit one form per call and you cannot remove a form submission. 2. Select a call history record 3. Choose 'Select conversation form' toward the bottom of the call history. 4. Choose a form name to submit 5. Fill out the form fields and click 'submit' at the top right corner.
Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the 'run history', which should show the run you just triggered as 'Succeeded'. In your Dynamics Sandbox Environment open up the 'contacts' in Dynamics CRM and open the contact 'Test Caller1'. Click on the 'phone call' in the contact timeline to expand the detail, notice that the phone call may have moved up in the timeline, since the record has been updated, and the activities are ordered by 'last updated' date-time. Within the phone call record on the contact timeline, you can observe that the call description, for the call you added a form response to, has been edited with an additional line beginning "Form submitted". The full form data that was submitted is populated on the call record.

	Fully open up the call record by clicking the ‘open record’ icon.  Observe the details of the call record. In the call records timeline The full form details have been submitted as a new note.
Notes	By default in Dynamics, this screen does not show “notes records” added to a phone call. You can customize this form in Dynamics admin to show notes added to phone calls. This is a cleaner way to store form responses and is a suggested practice. The customization can be seen below where a ‘timeline’ object is added to the phone call form. 

Alternative Test Method: Testing the Spoke webhook integrations with Postman

In some cases, you will want to test the integrations without using Spoke Phone, for example:

- Spoke Phone may already be live and you would not want the new flows to begin triggering on all events.
- Spoke Phone may not yet be configured or installed at the organization.

Postman can be used to simulate events that Spoke Phone triggers. Spoke Phone has included a Postman collection, to import and use to simulate Spoke Phone events and test your power automate flows.

Getting started with Postman

Firstly, it is recommended to install the desktop version of Postman. This supports 'pre-scripting' and variable declaration, making it much easier to work with the test scripts.

- Importing Postman collection
    - The sample Postman collection is easily imported using the 'Import' button located in the Postman toolbar.
- Important Notes:
    - The recommended test script below is carefully ordered so that the functionality of a real-world scenario is properly observed. The order is important to set up the appropriate data for the subsequent scripts. Only run the Postman tests individually or out of order once you understand the impact of doing so.
    - Updating Post URLs
        - In the test scripts below, it is important to ensure you update the post URLs as described in the test scripts, to post to ensure the test functions as expected.
    - Updating variables
        - In the test scripts below, it is important to ensure you update the variables correctly in the order described in the test scripts, to post to ensure the test functions as expected.
    - Re-running tests
        - You can re-run tests, however, the behavior of the system will not simulate real world Spoke Phone events unless you carefully edit the body content in order to simulate real-world scenarios. This involves editing the id's, dates and data content to carefully create the desired scenarios.

Running the tests with Postman: Recommended test script

NOTE: before running these tests, ensure you have completed the steps in section ”Importing the Power Automate Solution”

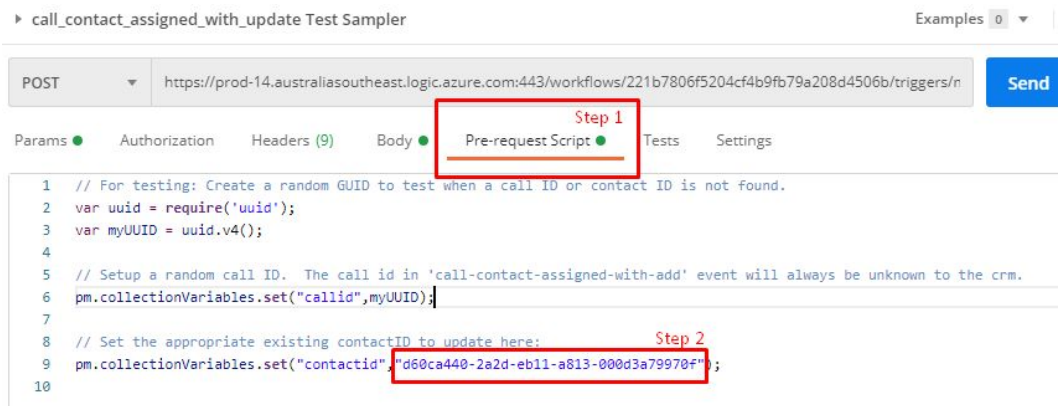
Test 1: Add new contact to CRM

When a Spoke Phone user receives a call from a number that does not exist in the Spoke Phonebook (and therefore the CRM), the Spoke Phone user has the option to ‘Add to CRM’ or ‘Select an existing contact to associate the number with’. This test runs the ‘add new contact to CRM’ option.

Step	
Setup	In the Power Automate solution, edit the flow “SpokePhone: SpokeWebhook - Contact assigned to Dynamics365”. Copy the URL of the first node in the flow. You may close the flow without saving. Open up “call_contact_assigned_with_add Test Sampler” in the Postman collection. Paste the URL copied from power automate into the ‘POST’ section of postman. There are no variables to update in this flow.
Run the test	Click ‘Save’ and then ‘Send’ in Postman. Observe a ‘202-accepted’ in the bottom section of Postman.
Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the ‘run history’, which should show the run you just triggered as ‘Succeeded’. You can open that run history and observe the flow state, variables and data as it ran through the steps. In your Dynamics Sandbox Environment (that you connected the flows to when importing the solution and saving the flows) you should now be able to open up the ‘contacts’ in Dynamics CRM and observe the newly added contact ‘Test Caller2’. Open up that contact and observe the detail added to the contact. Also observe a ‘phone call’ record added to that contact. Open up the phone call record and observe the detail added.
Notes	NOTE: your dynamics ‘contact’ and ‘phone call’ forms can be customized and may not show all fields supported in the Power Automate flows.

Test 2: Add unknown number to existing contact in CRM

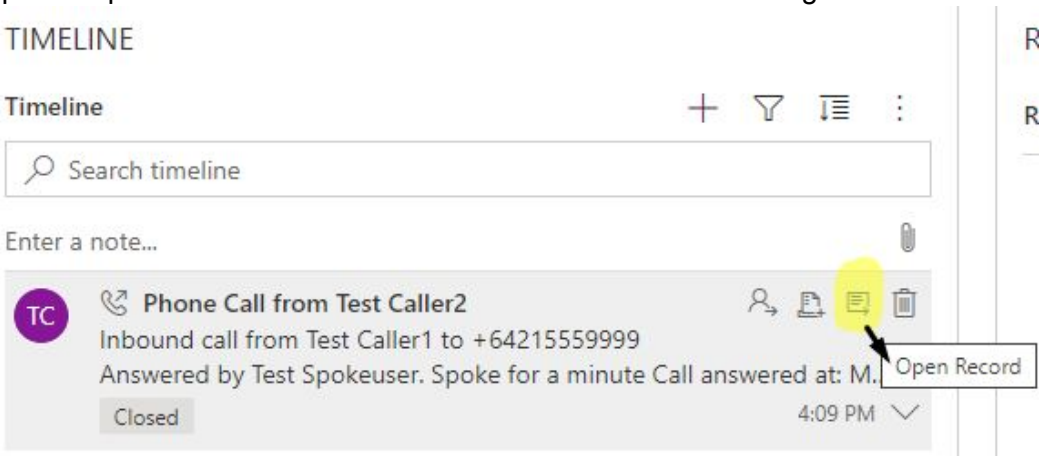
When a Spoke Phone user receives a call from a number that does not exist in the Spoke Phonebook (and therefore the CRM), the Spoke Phone user has the option to 'Add to CRM' or 'Select an existing contact to associate the number with'. This test runs the 'add new number to existing contact in CRM' option.

Step	
Setup	1. In the Power Automate solution, edit the flow "SpokePhone: SpokeWebhook - Contact assigned to Dynamics365". Copy the URL of the first node in the flow. You may close the flow without saving. 2. Open up "call_contact_assigned_with_update Test Sampler" in the Postman collection. 3. Paste the URL copied from power automate into the 'POST' section of postman. 4. Get the newly created dynamics contactid. To do this, open up the contact in dynamics and find the 'id' parameter in the URL looking something like: "….253A%252050&pagetype=entityrecord&etn=contact&**id=2f0c8308-b12e-eb11-a813-000d3a79a22c**" The 'red' highlighted section above is the id we need. The order of the url variables may not be the same as the above example. Note: Save this somewhere useful, as it will be reused again in next tests. 5. Head to the 'pre-request script' tab in Postman. (step 1 in screenshot below) 6. Edit the 'contactid' to be the contact id of the newly created dynamics contact. (step 2 in screenshot below) 
Run the test	Click 'Save' and then 'Send' in Postman. Observe a '202-accepted' in the bottom section of Postman.

Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the ‘run history’, which should show the run you just triggered as ‘Succeeded’. You can open that run history and observe the flow state, variables and data as it ran through the steps. Note that this time, under the “Add contact to CRM or update…” node (at the bottom of the flow) the flow went down ‘Case 2’ in the flow control, running the update routine, as opposed to add. In your Dynamics Sandbox Environment open up the ‘contacts’ in Dynamics CRM and open the contact ‘Test Caller2’. You should now see the new number added to the ‘Business Phone’ Field. Rerun the test to show the ‘new number’ getting added to the other phone fields. See the notes below for more detail.
Notes	The new unknown number has been added to the contact and Spoke Phone will now recognize it (the contact integration updates the Spoke phonebook with the new data). You can rerun this test to show when a new number is added to an existing contact, it adds that new number to the first empty field it finds in this order: mobile, business, home, telephone3. Note that if telephone3 contains a number already, the flow is set up to update it with the new number, since the logic is that the new number is more useful. You could edit the ‘body’ data of the postman to change the unknown phonenumber. The ‘data.call.assignedContact.phoneNumbers’ field contains the data. The ‘new number’ is the object that does not contain a ‘label’ property.

Test 3: Add a call log for a known caller

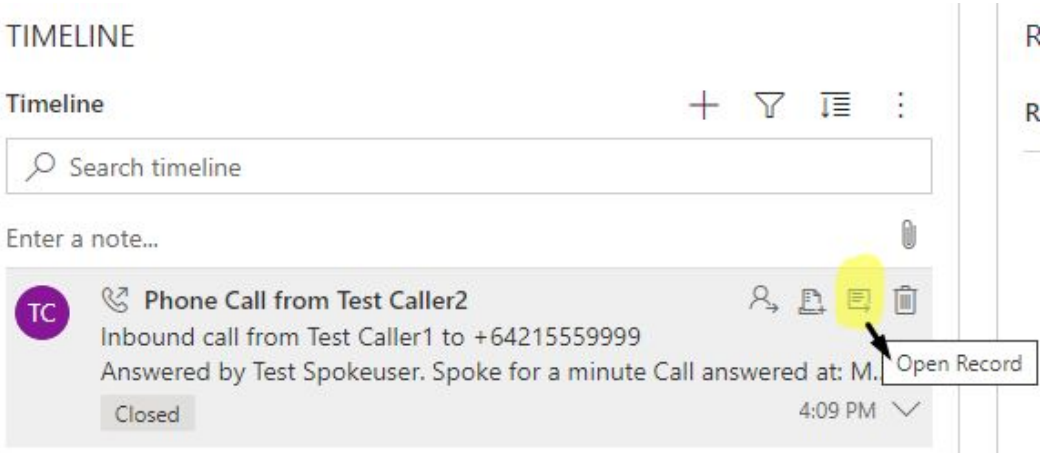
When a call is received from a number that is found in the Spoke phonebook or a call is made to a contact in the Spoke phonebook, when that call ends the information is written to contact that was matched. The call is logged against the CRM contact and the CRM 'user' that matches the Spoke Phone users email address that made or received the call. If the email is not found, the default CRM user is used (as configured in the flow installation guide)

Step	
Setup	1. In the Power Automate solution, edit the flow "SpokePhone: SpokeWebhook - Call Ended to Dynamics365". Copy the URL of the first node in the flow. You may close the flow without saving. 2. Open up "call_ended Test Sampler" in the Postman collection. 3. Paste the URL copied from power automate into the 'POST' section of postman. 4. Get the dynamics contactid for 'Test Caller2' as per steps in test2. 5. Open the 'pre-request script' section in Postman and set the 'contactid' variable to the contactid for 'test caller'. See steps in test2 for more detail if required. *Note: The callid will be dynamically generated to simulate a new call.*
Run the test	Click 'Save' and then 'Send' in Postman. Observe a '202-accepted' in the bottom section of Postman.
Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the 'run history', which should show the run you just triggered as 'Succeeded'. In your Dynamics Sandbox Environment open up the 'contacts' in Dynamics CRM and open the contact 'Test Caller2'. You should see the 'phone call' added to the timeline. Click on this to observe more text in the description field. Open the phone call event to view the detail added to the call log: 

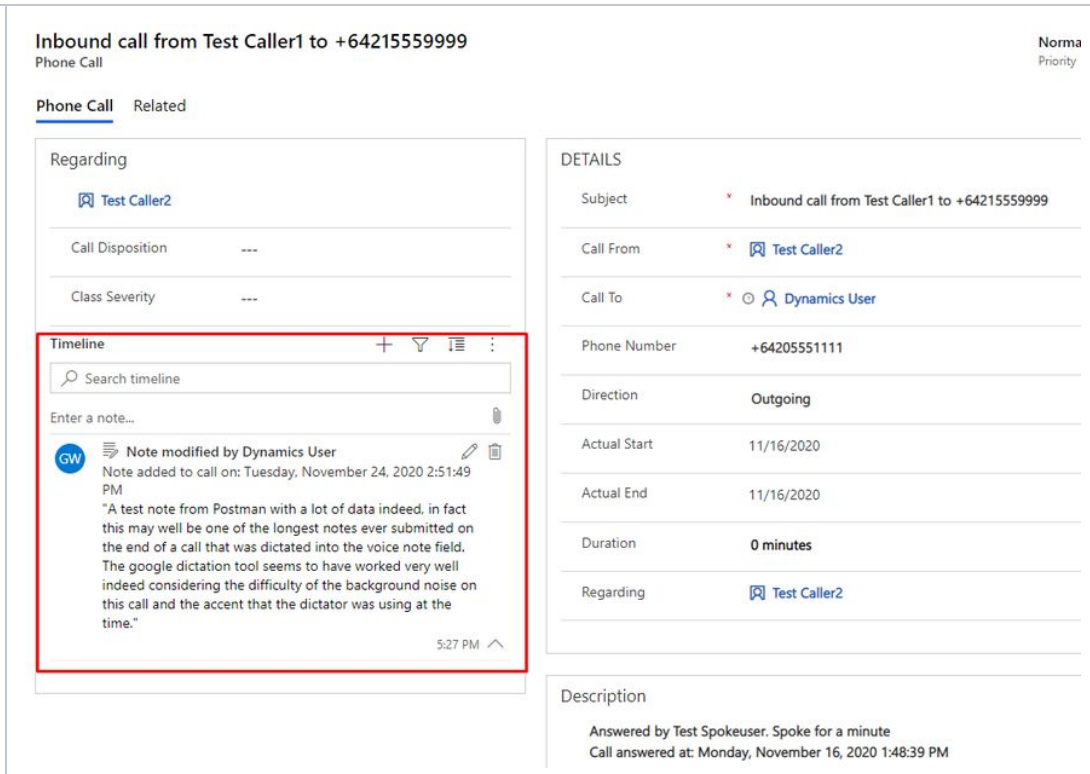
Notes	The date in the timeline shows when that ‘phone call’ event was last edited, not the actual date of the call. The date of the call is stored in the call description, as well as in the ‘due date’ and the ‘actual start’ and ‘actual end’ datetime fields. You can edit the datetimes in the body of the Postman to test different scenarios. You can run this test as many times as you like, each run will add a new call log. You can set the ‘callid’ parameter to be the same as an existing phone call record in order to test the exception case where the call already exists. The flow handles this by checking if the callid already exists and skips processing if it does.

Test 4: Add a note to a call

In the Spoke Phone app, the user can add a ‘voice note’ to a call. This is done post call from the call history screen. When a voice note is added, the webhook event ‘Call note created’ is triggered. Many voice notes can be added to the same call. The note is added to the phone call event in the CRM.

Step	
Setup	1. In the Power Automate solution, edit the flow “SpokePhone: SpokeWebhook - Call note added to Dynamics365”. Copy the URL of the first node in the flow. You may close the flow without saving. 2. Open up “call_note_added Test Sampler” in the Postman collection. 3. Paste the URL copied from power automate into the ‘POST’ section of postman. 4. Get the dynamics contactid for ‘Test Caller2’ as per steps in test2. 5. Also get the ‘call-id’ of a phone call you have added to this contact. Open a call on ‘Test Caller2’s timeline:  6. Find the ‘id’ parameter in the URL looking something like: ”2520%253A%252050&pagetype=entityrecord&etn=phonecall&**id=00515f3d-3d56-4c20-9234-c6f894a7e9b7**” The ‘red’ highlighted section above is the id we need. The order of the url variable may not be the same as the above example. Note: Save this somewhere useful, as it will be reused again in next tests. 7. Open the ‘pre-request script' section in Postman and set the ‘contactid’ variable to the contactid for ‘test caller2’. See steps in test2 for more detail if required. 8. Set the ‘callid’ variable to the call id for one of the phone call records on ‘test caller2’.

Run the test	Click ‘Save’ and then ‘Send’ in Postman. Observe a ‘202-accepted’ in the bottom section of Postman.
Validate	In Microsoft Power Automate, go to the details page of the flow. Observe the ‘run history’, which should show the run you just triggered as ‘Succeeded’. In your Dynamics Sandbox Environment open up the ‘contacts’ in Dynamics CRM and open the contact ‘Test Caller2’. On the contact timeline you can observe the note added as a new ‘note’ activity. The full voice note that was dictated (or typed) is populated on the call record. Click on the ‘phone call’ in the timeline to expand the detail, notice that the phone call may have moved up in the timeline, since the record has been updated, and the activities are ordered by ‘last updated’ date time. In the phone call detail, you can observe the description is expanded with a comment “Note added to call on: {{datetime}}”. This makes reference that this call had a note added to it. Fully open up the call record by clicking the ‘open record’ icon (as noted in test 3). Observe the details of the call record.
Notes	By Dynamics default this screen does not show notes added to a phone call. You can customize this form in Dynamics admin to show notes added to phone calls. This is a cleaner way to store notes and is a suggested practice. The customization can be seen below where a ‘timeline’ object is added to the phone call form. The flow adds this by default and when you make the customization it will show up here, at which point you may choose to remove the part of the flow that adds the note to the contacts timeline.


Inbound call from Test Caller1 to +64215559999
Phone Call
Norma
Priority
Phone Call
Related
Regarding
Test Caller2
Call Disposition

Class Severity

Timeline
Search timeline
Enter a note...
GW
Note modified by Dynamics User
Note added to call on: Tuesday, November 24, 2020 2:51:49 PM
"A test note from Postman with a lot of data indeed, in fact this may well be one of the longest notes ever submitted on the end of a call that was dictated into the voice note field. The google dictation tool seems to have worked very well indeed considering the difficulty of the background noise on this call and the accent that the dictator was using at the time."
5:27 PM
DETAILS
Subject
Inbound call from Test Caller1 to +64215559999
Call From
Test Caller2
Call To
Dynamics User
Phone Number
+64205551111
Direction
Outgoing
Actual Start
11/16/2020
Actual End
11/16/2020
Duration
0 minutes
Regarding
Test Caller2
Description
Answered by Test Spokeuser. Spoke for a minute
Call answered at: Monday, November 16, 2020 1:48:39 PM

Troubleshooting

“Trigger is broken”: If you see something wrong with the trigger on a Spoke Phone webhook event, this can be resolved by relaxing the triggers ‘schema’. To find the issue, you can download the event data that failed from the Spoke Phone Account Portal, and validate it against the Power Automate Flow ‘trigger’ schema. There are many websites that offer JSON schema validation online. The result of this should tell you where the trigger error is to resolve the issue in the schema.

Updating and customizing the flows.

Spoke Phone intends the collection of sample flows and guide for integration with Dynamics 365 using Power Automate to cover the most pertinent needs in a generic format.

You will need to customize these flows to suit your business and purposes, and in some cases to make them functional within your environment.

The following are some ideas on what extensions could be built to these flows:

Probable extension cases:

- Store call highlights using the `'call.highlight.recording_available'` event:
  Highlights are added within a call to record a 30 second snippet of sound and tag that snippet with a certain keyword. The webhook will notify of a highlight audio being ready to be downloaded. The likely use case, pending security and privacy reviews, would be to store these recorded snippets into the CRM against the contact.

- Store call recordings using the `'call.recording.available'` event:
  Entire call recordings are available in the Spoke Phone App, however you may want to download those to store somewhere locally. Pending security and privacy reviews, and the storage limitations of your CRM environment, you may want to download and store them in the CRM.

- Voicemail transcription `'call.voicemail.available'` event:
  You could download the voicemail and store it against the missed call in the CRM, or perhaps just add a new note with the transcribed text into the CRM against the call record.

- Saving call log data into the CRM even if there is no assigned contact
  - Perhaps saving it into a temporary 'unknown' contact, or just relating the call record to something other than contact and creating the appropriate report.
  - When contact is assigned (add or update) the call log can then be updated to 'attach' it to the appropriate contact.

Other possible use cases:

- Custom contact fields - Update custom fields with form responses.
- Create a follow-up activity if a voicemail is received. Automatically close if a call is received or made in the meantime.
- Creating actionable activity, for example, a CRM case, when a form is submitted and certain responses are collected.
- Sending form data to excel for rapid analytics of form data collected: When complex forms with many nested objects and arrays are created, the included sample flow will be expensive (in terms of billable actions). This can be solved by implementing an API call to flatten a dynamic JSON object into an array of name:value pairs to send to Excel.