

Integration Tools and Utilities for Alchemer Workflow

Overview

Alchemer Tools is a set of built-in Alchemer utilities that perform data transformation and output actions within a workflow. Organizations use Tools to manipulate data, construct messages, run custom logic, and send outbound requests — all without leaving the Alchemer workflow environment.

The Alchemer Tools integration supports data creation, transformation, and delivery actions natively within workflow. Tools allows Alchemer to define values, build messages, execute custom JavaScript, send outbound webhooks, and parse date and timestamp data — enabling complex workflow logic and downstream integrations without manual intervention.

Common uses for Alchemer Tools in workflow

- Hard-code or dynamically set values for use as merge codes in downstream workflow steps
- Run custom JavaScript to transform data and return results as merge codes
- Build plain text messages using workflow merge codes
- Send outbound webhook requests to external systems with workflow data
- Parse date and timestamp data to extract formatted date details and calculations

What can Alchemer Tools do?

- [Create hidden values](#)
- [Run custom JavaScript](#)
- [Construct message](#)
- [Outbound webhook](#)
- [Get date/timestamp details](#)
- [Get current timestamp](#)

You will need

- An Alchemer plan that includes integrations and the Integration Manager permission enabled.
 - [Contact us](#) if you are unsure if your plan includes integrations.

Setup Alchemer Tools in workflow

[Tools | Create hidden values](#)

Use this action to define key/value pairs that will be available as merge codes in subsequent workflow steps. Values can be hard-coded or built dynamically using merge codes from previous steps.

You will need:

- The key names and values you want to define

Configure the action

1. Open your workflow in **Workflow Builder**.
2. On the right side, drag and drop the **Tools** connection where you want the action to trigger.
3. In the connection box, click the pencil icon in the top right corner.
4. Select **Tools | Create hidden values**.
5. **Tools | Select Alchemer fields:** Select any existing Alchemer workflow fields you want to make available for use in this action.
6. **Tools | Define values — Hidden Values:** Add one or more key/value pairs. Enter a key name (e.g. *email*), then enter the value. You can paste merge codes from previous workflow steps directly into the value field to build values dynamically.
7. **Tools | Define values — Merge codes (optional):** Expand to view the list of available merge codes based on the fields you have selected. These can be copied and pasted into value fields.
8. Save the action.

Status codes

- 200: Hidden values successfully created and available as merge codes
- 400: The action returned an error

Tools | Run custom JavaScript

Use this action to write and execute custom JavaScript within a workflow step. The script must return an object with key/value pairs — each key in the returned object becomes available as a merge code in subsequent workflow steps.

You will need:

- The JavaScript you want to run
- The variable names you want returned as merge codes

Configure the action

1. Open your workflow in **Workflow Builder**.
2. On the right side, drag and drop the **Tools** connection where you want the action to trigger.
3. In the connection box, click the pencil icon in the top right corner.

4. **Select Tools | Run custom JavaScript.**
5. **Tools | Define variables:** Define the input variables your script will use. These variables can be populated using merge codes from previous workflow steps.
6. **Tools | Fields to use:** Select any Alchemer workflow fields you want to make available to the script.
7. **Tools | Construct script — Script:** Write your JavaScript in the script input. Your script must return an object with key/value pairs using the defined variables as keys. Each key in the returned object will be available as a merge code in subsequent workflow steps.
8. **Tools | Construct script — Merge codes (optional):** Expand to view available merge codes. These can be added to the script using the **Add Property To Merge code list** button.
9. Save the action.

Status codes

- 200: Script executed successfully and output returned as merge codes
- 400: Script returned an error or failed to execute

Tools | Construct message

Use this action to build a plain text message using merge codes from the workflow. The constructed message is returned as a merge code available in subsequent steps.

You will need:

- The merge codes and text content you want to include in the message
- To format your message the way you want using the mapper and builder

Configure the action

1. Open your workflow in **Workflow Builder**.
2. On the right side, drag and drop the **Tools** connection where you want the action to trigger.
3. In the connection box, click the pencil icon in the top right corner.
4. Select **Tools | Construct message**.
5. **Tools | Fields to push:** Select the Alchemer workflow fields you want to make available for use in your message.
6. **Tools | Construct message:** Build your plain text message using the mapper and builder. Insert merge codes from the workflow to include dynamic values in the message.
7. Save the action.

Status codes

- 200: Message successfully constructed and available as a merge code
 - 400: The action returned an error
-

Tools | Outbound webhook

Use this action to send an HTTP POST or GET request to an external endpoint from within a workflow. Workflow field values can be passed as field parameter codes in the URL and request body.

You will need:

- The URL of the external endpoint you want to send data to
- The workflow fields you want to include in the request

Configure the action

1. Open your workflow in **Workflow Builder**.
2. On the right side, drag and drop the **Tools** connection where you want the action to trigger.
3. In the connection box, click the pencil icon in the top right corner.
4. Select **Tools | Outbound webhook**.
5. **Tools | Fields to push**: Select the Alchemer workflow fields whose values you want to send in the webhook request.
6. **Tools | Webhook setup — Method**: Select the HTTP method for the request. Defaults to **POST**.
7. **Tools | Webhook setup — Custom Headers (optional)**: Add any HTTP header key/value pairs required by the receiving endpoint.
8. **Tools | Webhook setup — URL Endpoint**: Enter the URL of the external endpoint that will receive the request.
9. **Tools | Webhook setup — Body type**: Select the format for the request body. Defaults to **JSON**.
10. **Tools | Webhook setup — Construct your body**: Build the request body. Use the **Add Property To Construct your body** button to map fields into the body.
11. **Tools | Webhook setup — Field parameter codes (optional)**: Field parameter codes can be inserted into the webhook URL and body. These codes are automatically replaced with the corresponding workflow field values when the workflow runs. Expand to view available codes based on the fields you selected.
12. Save the action.

Status codes

- 200: Outbound webhook successfully sent
 - 400: The request was rejected or the endpoint returned an error
-

Tools | Get date/timestamp details

Use this action to parse a date or timestamp field from your workflow and extract formatted date details and calculations. The parsed values are returned as merge codes available in subsequent workflow steps.

You will need:

- A workflow field containing a date or timestamp value to parse

Configure the action

1. Open your workflow in **Workflow Builder**.
2. On the right side, drag and drop the **Tools** connection where you want the action to trigger.
3. In the connection box, click the pencil icon in the top right corner.
4. Select **Tools | Get date/timestamp details**.
5. **Tools | Parse date/timestamp**: Select the workflow field containing the date or timestamp you want to parse.
6. **Tools | Select options — Date/timestamp fields**: Map the Alchemer field containing your date or timestamp as the input. A comparison date/timestamp field is optional.
7. **Tools | Select options — Default input time zone**: Select the time zone to apply when parsing input date/timestamps that do not include time zone information. Defaults to **America/New_York**.
8. **Tools | Select options — Output time zone**: Select the time zone for the output. All date and time details returned will reflect this time zone. Defaults to **America/New_York**.
9. **Tools | Select options — Week starts on**: Select the day your week begins. Used for week-based date calculations. Defaults to **Sunday**.
10. **Tools | Select options — Weekend day 1 / Weekend day 2**: Select the days that define your weekend. Used for business day calculations. Default to **Saturday** and **Sunday**.
11. **Tools | Select options — Custom format (optional)**: Enter a date/time formatting string in [Moment.js format](#) to customize the output format (e.g. `dddd, MMMM Do YYYY, h:mm:ss a`).
12. **Tools | Calculations**: Select the date calculations you want performed and returned as merge codes (e.g. days between dates, business days, day of week).
13. **Tools | Get data back**: Select which timestamp data you want to get back as merge codes in

your Workflow.

14. Save the action.

Status codes

- 200: Date/timestamp successfully parsed and details returned as merge codes
 - 400: The action returned an error or the date/timestamp field could not be parsed
-

Tools | Get current timestamp

Use this action to capture the current timestamp at the moment the workflow runs and return it as merge codes available in subsequent workflow steps. The timestamp can be returned in multiple formats, including UTC, parsed, ISO, and a custom format.

You will need:

- No prior workflow fields are required — the timestamp is generated at runtime

Configure the action

1. Open your workflow in **Workflow Builder**.
2. On the right side, drag and drop the **Tools** connection where you want the action to trigger.
3. In the connection box, click the pencil icon in the top right corner.
4. Select **Tools | Get current timestamp**.
5. **Tools | Select options:** Select the output time zone. Optionally, enter a custom date/time format in **Moment.js format** (e.g. `dddd, MMMM Do YYYY, h:mm:ss a`).
6. **Tools | Get data back:** Select which fields you want to get back as merge codes (current timestamp in UTC, parsed, ISO, custom).
7. Save the action.

Status codes

- 200: Current timestamp successfully captured and returned as merge codes
 - 400: The action returned an error
-

Testing and Troubleshooting

Testing and Validation

How to test

- Trigger the workflow and monitor individual runs in the **Monitor** tab.
 - Click on individual workflow runs to see metadata outputs for each Tools step.

- Confirm that the expected values, messages, or date details are returned and available as merge codes in subsequent steps.
- Use metadata for verification and debugging.

How to verify results

- Add a send email action to yourself with the output merge codes included to confirm returned values are correct.
 - Ensure retrieved or constructed values match expectations.
-

Monitoring Integration Activity

Where to find logs

- Go to Results → Monitor.
- Select the Tools step you want to inspect.

What logs display

- Input/Output
-

Troubleshooting

Configuration issues

- Missing required fields (e.g. Hidden Values, URL Endpoint, date/timestamp input)
- Merge codes from previous steps not yet available at the point the Tools action runs

Script errors (Run custom JavaScript)

- Script does not return an object with key/value pairs
- Variable names do not match the defined variables from the previous step
- JavaScript syntax errors

Outbound webhook issues

- Invalid or unreachable URL endpoint
- Incorrect body type or malformed request body
- Missing required custom headers

Date/timestamp issues

- Date/timestamp field value is empty or in an unrecognized format
 - Incorrect time zone selected for input or output
 - Custom format string not valid Moment.js syntax
-

FAQs

What permissions do I need?

Integration Manager in Alchemer.

When does the action run?

When the workflow triggers and reaches the Tools step.

Can I use multiple Tools actions in one workflow?

Yes. Tools actions can work independently or together, and their output merge codes can be passed into subsequent steps.

Can I use merge codes from a Tools step in later workflow steps?

Yes. Each Tools action returns its output as merge codes that are available to all subsequent steps in the workflow.

Why isn't my data appearing in downstream steps?

Check the Monitor tab for configuration errors, missing required fields, or output mapping issues.

What if I need additional functionality?

Contact Alchemer Support for enhancement requests.

Related Articles
