

Onboarding - Automation Triggers and Actions

Last Modified on 10/02/2026 3:07 pm CDT

What is an Automation?

An automation is a directed graph of decision points and actions that runs when a specific event happens (for example, the automatic sending of an email once a task has been completed by a candidate). You define the trigger (e.g., "when a task's status changes"), and the automation walks through the graph by evaluating conditions, taking actions, and pausing at time nodes, all within a single execution path.

The screenshot displays a visual automation builder interface with four main configuration panels arranged horizontally:

- Placement Events:** Contains fields for 'Automation Name' and 'Email Notification'. The description states: 'This sends an email when a placement is created'. It includes a 'Dependencies' section with a 'Status' dropdown and a checkbox for 'Run on new placement creation'. The 'Run Strategy' is set to 'Allow returns'.
- condition_1:** Shows a rule configuration where 'Any of the following must be true'. A single condition is defined: 'Name' 'does not contain' 'Candidacy'. A note below states: 'This branch runs when Name does not contain "Candidacy" (case sensitive)'. There are buttons to 'Add condition' and 'Add group'.
- action_2:** Configured for 'Create Onboarding Link'. Parameters include 'Onboarding Link Type' (Placement), 'Assignee Type' (Employee), 'Locale' (Auto), and 'Redirect To'. It also has a 'Theme' dropdown set to 'Select theme' and a 'Require OTP Verification' dropdown set to 'No'. A note mentions: 'When enabled, the employee must verify their identity via a one-time code before accessing the onboarding form'.
- action_1:** Configured for 'Send Email'. Parameters include 'Email service' (Onboarded Email Service), 'Email template' (Onboarded - Content), 'Theme' (No theme), and 'Send email to' (Employee). The 'Email subject' is 'New Onboarded Notification'. The 'Email body' contains a paragraph with a dynamic link: 'Hi {{ employee.first_name }} {{ employee.last_name }}. Please complete the forms in the onboarding link below. [Onboarding Link](#). Thank you, Your Onboarding Team'. There is a 'Dynamic Template Data' section at the bottom.

Without automations, you would need to manually create tasks, send emails, and monitor expiration dates. With Automations, those events (conditional task creation, link generation, reminders, compliance workflows) run consistently every time that event occurs.

Note You will need to be a member of the "Admins" User Group to create and manage Automations.

For more information on permissions, please see the article titled [Onboarding - User Groups and Permissions](#).

This article lists every Trigger Entity Type and Dependency Property, the values each property accepts when you reference it in a condition rule, every action type with its parameters and outputs, and the time-delay baselines available in automations.

Note For more information on Automations, please see the following articles:

- [Onboarding - Understanding Automations](#)
- [Onboarding - How to Setup Automations](#)

Entity Types and Dependency Properties

An automation executes when an Entity Type ("Task or "Placement") experiences a change to one of its Dependency Properties. You can also configure an automation to fire when a record is created for that Entity Type.

Entity Type	Dependency Property	Executes when...	Additional Notes
task	status	Task "Status" changes (e.g. moves from open to submitted)	
task	expired_at	Task "Expiration Date" is set or changes	
task	due_at	Task "Due Date" is set or changes	
task	next_action	Task "Next Action" value changes	The "task.next_action" executes every time a task moves between stages in its workflow. Most use cases that need a notification when onboarding begins, like a welcome email, want "placement.created" instead.
placement	status	Placement "Status" changes	
placement	progress	Placement "Progress" field changes	

Note Selecting "Run on new record creation" will execute on either "task.created" or "placement.created":

Trigger Settings

Automations continuously monitor changes to certain properties of an entity. Those properties are called "dependencies". Changes to one or more dependencies will trigger this automation.

Entity Type ?

Select entity type

☐ Run on new record creation

Condition Values by Trigger Property

When you reference a trigger property in a condition rule, the value you type must match how that property is stored within the Onboarding Portal. The table below lists the accepted values for each property in the "Entity Type and Dependency Property" table above.

Field	Allowed Values	Notes
task.status	<ul style="list-style-type: none">requires_actioncompletedexpired	Closed set. Use with is / is not / is in / is not in.
task.next_action	<ul style="list-style-type: none">employee_actionemployer_actionsystem_action	Closed set. Describes who the task is currently waiting on.
task.due_at	Any ISO 8601 timestamp (e.g. 2026-06-15T17:00:00Z), or null when no due date is set	Stored as a timestamp, not a string the rule engine can compare with greater than / less than. Use exists / does not exist to test whether a due date is set, and use a time node with the Task Due baseline for "X days before due."
task.expired_at	Any ISO 8601 timestamp (e.g. 2026-12-31T23:59:59Z), or null when the task never expires	Same comparison limits as due_at. Use exists / does not exist and the Task Expired time node baseline.

Field	Allowed Values	Notes
task.assignee	Sub-paths: <ul style="list-style-type: none"> task.assignee.email task.assignee.uid task.assignee.is_admin 	Pick the sub-path in the picker. task.assignee on its own is the whole object and won't compare cleanly. Use exists on task.assignee to test whether an assignee is set.
placement.status	<ul style="list-style-type: none"> completed incompleted 	Derived from placement.progress: completed when progress hits 1.0, otherwise incompleted.
placement.progress	<ul style="list-style-type: none"> 0.0 0.1 0.2 ... 1.0 	Stored as a one-decimal string, not a number. is / is not work (e.g. is 1.0 for "fully complete"). Numeric greater than / less than won't reliably evaluate against a string value — use placement.status for "is the placement done?" instead.

Note Values are case-sensitive and the input is always plain text. "Completed" (capital "C") won't match "completed" (lowercase), and "employer-action" (with a hyphen) won't match "employer_action" (with an underscore). Type the value exactly as it appears in the table above.

Action Types

Each action performs a specific operation. Parameters marked optional can be omitted depending on the action's context.

Outputs are available to later nodes in the same automation using `{{action_name.output_field}}` syntax.

Action	What it does	Key Parameters	Outputs Available to Later Nodes
--------	--------------	----------------	----------------------------------

Action	What it does	Key Parameters	Outputs Available to Later Nodes
Send Email	Sends an email through the Onboarding Portal's email service, or through a connected SendGrid account	<ul style="list-style-type: none"> email_service email_subject email_body recipient_type (employee or custom) recipient_email, theme_id (optional) sendgrid_template_id (optional) dynamic_template_fields (optional) 	recipient_email
HTTP Request	Calls an external endpoint	<ul style="list-style-type: none"> url method (GET, POST, PUT, PATCH, DELETE) headers (optional) url_parameters (optional) body (optional) 	<ul style="list-style-type: none"> success, status_code response_body (optional) request_body (optional)
Create Task	Creates a task on the placement (de-duplicates by lineage)	task_lineage_uid	<ul style="list-style-type: none"> task_id task_status form_name form_id
Create Compliance Requirement	Creates a suggested task or compliance requirement; optionally also creates the task	<ul style="list-style-type: none"> task_lineage_uid also_create_task (optional) 	<ul style="list-style-type: none"> task_ids form_name form_id placement_ids
Assign Task	Assigns a task to a user or group (task triggers only)	<ul style="list-style-type: none"> assignment_type (individual_user or group) assignee_id or group_id 	<ul style="list-style-type: none"> assignee_email assignee_id assignment_type group_name (optional)

Action	What it does	Key Parameters	Outputs Available to Later Nodes
Create Onboarding Link	Generates a fresh signed onboarding link covering current tasks	<ul style="list-style-type: none"> onboarding_link_type (employee or task) assignee_type (employee or employer) locale, redirect_to (optional) active_theme_id (optional, pass a theme UID to override the employer/account default. otp_required (optional) otp_methods (optional) 	onboarding_link
Set Task Expiration	Sets or updates the expiration timestamp for a task on the placement	<ul style="list-style-type: none"> task_lineage_uid expiration_strategy (specific_time, relative_time, or form_field) expiration_time_amount (optional) set_earliest_expiration_time (optional) 	<ul style="list-style-type: none"> task_uid expiration_time

Note The "Set Task Expiration" action may require additional setup before becoming available as an option when creating an automation. Please contact TempWorks Support for more information.

Time Node Baselines

A Time Node introduces a delay before the automation proceeds. The baseline determines what moment in time the delay is measured from.

Reference Time	UI Label	Wait until...
now	Now	Trigger time, plus the offset

Reference Time	UI Label	Wait until...
task_expired	Task Expired	The task's expiration timestamp, plus the offset
task_due	Task Due	The task's due timestamp, plus the offset

Note Please keep in mind the following with Time Node baselines:

- The offset field is labeled "After / Before".
- Time units are "Minute", "Hour", and "Day" (capitalized).
- Use a negative number to run before the baseline timestamp.
- The "When data changes during wait" toggle determines whether the automation re-evaluates conditions using current data if the specified task or placement is updated during the wait period. Options are "Use the latest data from Onboarded (true)" or "Use data from before the wait time (false)":

The screenshot shows a configuration window for a timing node named 'timing_1'. The main setting is 'Continue after 2 minutes'. Below this, there are three fields: 'Reference time' set to 'Now', 'After / Before' set to '2', and 'Time Unit' set to 'Minute'. At the bottom, there is a section titled 'When data changes during wait:' with a dropdown menu currently set to 'Use data from before the wait time'. This dropdown is highlighted with a red rectangular box.

Rerun Behavior Options

Rerun Strategy controls what happens when the trigger fires multiple times.

The Rerun Strategy can be configured either when initially creating an automation or within the Start "Trigger" Node.

You can select from the following options:

- **Allow Reruns:** Every trigger fires the automation independently. If the trigger fires twice, the automation runs twice, even if the first run is still in progress.
- **Do Not Allow Reruns:** The trigger only fires the automation if no other execution is currently active.

Subsequent triggers are ignored.

- **Allow Reruns and Cancel Previous Runs:** A new trigger cancels any in-progress execution and starts fresh.

The screenshot shows the configuration for a 'Placement Events' automation. The 'Automation Name' is 'Email Notification' and the 'Description' is 'This sends an email when a placement is created'. The 'Dependencies' section shows 'Status' as a dependency. The 'Run on new placement creation' checkbox is checked. The 'Rerun Strategy' dropdown is highlighted with a red box, showing the following options:

- Allow reruns
- Allow reruns and cancel previous runs
- ✓ Allow reruns (selected)
- Do not allow reruns

Variable Name Spacing in Templates

Action outputs are addressable in templates using `{{action_name.output_field}}`. The action name in the automation builder becomes the namespace.

For example, if you name a "Send Email" action as "Welcome email", its output is `{{Welcome email.recipient_email}}`.

Related Articles
