

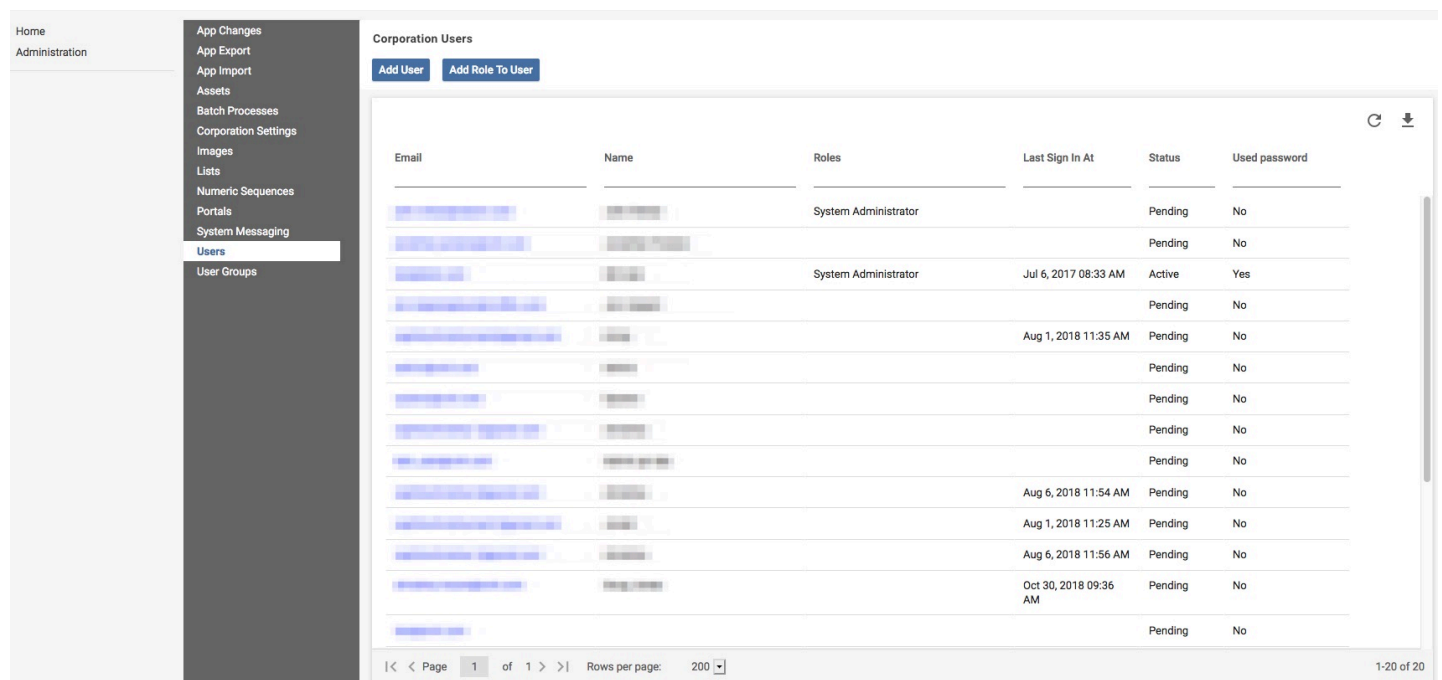
Automating User Onboarding

This tutorial provides tips and best practices for automating various user actions (such as onboarding). While all of the actions described in this tutorial can be performed manually, there are obvious benefits to automating these tasks.

Understanding Onit User Objects

In most Onit environments, each user has *two* different (yet related) user objects:

1. **Users:** This object is visible (to Onit System Administrators) on an environment's **/admin** page (as seen below). It provides the user with the ability to actually log into the environment and it tracks a minimal number of properties (e.g., the user's name, email address, last sign in timestamp - it is not possible to add new custom properties to this object).



Email	Name	Roles	Last Sign In At	Status	Used password
		System Administrator		Pending	No
				Pending	No
		System Administrator	Jul 6, 2017 08:33 AM	Active	Yes
				Pending	No
			Aug 1, 2018 11:35 AM	Pending	No
				Pending	No
				Pending	No
				Pending	No
				Pending	No
			Aug 6, 2018 11:54 AM	Pending	No
			Aug 1, 2018 11:25 AM	Pending	No
			Aug 6, 2018 11:56 AM	Pending	No
			Oct 30, 2018 09:36 AM	Pending	No
				Pending	No

2. **User Preferences Provider App:** In most cases, an app named **User Profiles** is created in the environment and it is designated as a User Preferences Provider ([as explained here](#)). Each corporation user object (described in the bullet-point above) will have exactly one matching Record in the **User Profiles** app. Since **User Profiles** is an Onit App, you can customize it however you like. For example, you can add new Fields to “decorate” users with custom properties. In addition, you can fire Business Rules, Actions, Buttons and other workflow-related items on these Records. As a result, you can essentially treat users like you would any other business object from a workflow standpoint.

[illegible]

It's important to understand how and when the above objects are *created* in Onit.

1. A corporation user object is created when *any* of the following events occur:
 - A new user is added as a Participant on *any* Record in the environment.
 - A System Administrator manually creates a user by clicking **Add User** on an environment's **/admin** page.
 - A user successfully authenticates into Onit via SSO for the first time and the environment's "auto create user" setting is enabled (this setting is described in more detail later in this tutorial).

Home
Administration
App Changes
App Export
App Import
Assets
Batch Processes
Corporation Settings
Images
Lists
Numeric Sequences
Portals
System Messaging
Users
User Groups

Corporation Users

Add User
Add Role To User

Email	Name	Roles	Last Sign In At	Status	Used password
		System Administrator		Pending	No
				Pending	No
		System Administrator	Jul 6, 2017 08:33 AM	Active	Yes
				Pending	No
			Aug 1, 2018 11:35 AM	Pending	No
				Pending	No
				Pending	No
				Pending	No
				Pending	No
			Aug 6, 2018 11:54 AM	Pending	No
			Aug 1, 2018 11:25 AM	Pending	No
			Aug 6, 2018 11:56 AM	Pending	No
			Oct 30, 2018 09:36 AM	Pending	No
				Pending	No

1
of 1
>

Rows per page:
200

1. A **User Profile** Record is created when *any* of the following events occur:

- A new corporation user object is created (regardless of how this object is created). When the corporation user object is created, Onit automatically checks the environment for a User Preferences Provider App. If this App exists, Onit checks the App for a Record that has a **name** Field value that matches the corporation user object's email address. If no match is found, Onit automatically creates a new Record in this app.
- A user, Onit Action, or API call creates the Record. In other words, since **User Profiles** is just an App, Records can be created in this environment just like any other app. This approach is often taken if the the user in question cannot log in via SSO (e.g., the client doesn't use SSO internally and/or the user doesn't work directly for the client).



It is critical to understand all of the above details before continuing on with this tutorial.

The "Auto Create User" SSO Setting

If an environment has single sign-on (SSO) enabled, as mentioned above you can *auto*-create their user object when they SSO into Onit for the first time.

To enable this feature, browse to your environment's **Administration** page and open the **Edit Corporation** pane. Navigate to the **Security** tab and check **Auto Create User**.

The screenshot shows the 'Corporation Settings' page with the 'Security' tab selected. The 'Authentication Strategy' is 'Single Sign On and Password'. The 'Auto Create User' checkbox is checked. Other settings include 'Idp Target URL', 'Idp logout URL', 'Idp Cert Fingerprint (SHA1)', 'Onit's SAML Issuer ID', 'Sign SAML Request' (unchecked), 'SSO Cert Type' (Self signed), 'SSO Type' (SAML), 'SSO login button text', 'Password login button text', and 'Session Timeout' (1).

Enabling this feature will ultimately create a new user Record in your User Preferences Provider App for each brand new SSO sign in. You can use this event (i.e., the creation of a new Record) as a trigger for automation, by firing Business Rules and Actions. For example, you could automatically add the user to a default group, send the user an email, start an approval process for the user, etc. Tips on how to build these types of automation are covered below.

Building and Executing a New User Approval Process

When onboarding a new user, you may want to send the user through an approval workflow before activation. Approvers on this workflow might include the user's supervisor or anyone else who needs to approve a new hire.

Once a user has been approved, they can be added as a Participant on their own **User Profiles** Record in order to customize their Onit preferences.

The following is an example of a new user approval process workflow:

1. A new user signs into the Onit environment through SSO.
2. A new user Record is created in the **User Preferences Provider** App.
3. The user is temporarily deactivated using the **Activate or Deactivate User** Action (explained below).
4. The user's supervisor is added as an approver on the Record.
5. Pre-approval actions are taken on the Record (e.g., assigning an employee number).
6. The new user is ultimately approved.
7. The new user is reactivated using the **Activate or Deactivate User** Action.
8. An onboarding email is sent to the user, welcome them to Onit.
9. The user is added as a Participant on their own Record.

The Activate or Deactivate User Action

The **Activate or Deactivate User** Action does exactly what its name implies. Deactivated users cannot log in to Onit.

The Action's **User Email** property (shown below) should contain a Liquid expression that evaluates to the email address of the user that you want to activate or deactivate.

Activate or Deactivate User

Name *	New Action
Description	
Status *	
User Email	<div>Editor</div>
Comment	Provide a brief description of the change you made (optional)
	<div>Save</div>

The Add to UserGroup and Remove from UserGroup Actions

The **Add to UserGroup** and **Remove from UserGroup** Actions also do actually what their names suggest. The former Action is often used to default users into a User Group whenever their user objects are automatically created (via an SSO login). Also, it can be helpful to tie both Actions to checkbox Fields in the **User Profiles** App, which allow clients to easily add/remove users from User Groups by simply checking/unchecking boxes (versus having to browse to the **/admin** page).



If you are not familiar with Onit User Groups, [see out tutorial on this topic.](#)

The Action's **User Email** properties should contain a Liquid expression that evaluates to the email address of the user that you want to add or remove from a User Group. The **User Group Name** property should be the plain text name of the User Group in question (e.g., Law Department).

Add User to UserGroup

Name *	New Action
Description	
Email *	<input type="text"/> Editor
User Group Name *	<input type="text"/> Editor
Comment	Provide a brief description of the change you made (optional)


Save

Adding a User as a Participant on Their Own Record

It is recommended that you always automatically add each user as a Participant on their own **User Profiles** record. There are numerous benefits to doing so. For example, this ensures that the user will always have a corporation user object. In addition, this allows the user to customize some aspects of their own preferences, such as their phone number or enabling their out of office delegate.

To add a user as a Participant on their own **User Profiles** record, an **Add Participant** Action should be used and it should be configured as shown below.

Add Participant

Name *	Add User as Participant	
Description		
Emails *	<code>{{user_name}}<{{name}}></code>	
Role *	User	
Liquid Role (Overrides Role)		Editor
Phase *	App Configured	
Comment	Provide a brief description of the change you made (optional)	

[Save](#)

The **Emails** property should evaluate to the user's email address. The user's name can optionally be included in angle brackets after the email – this is recommended but not required, the angle brackets will push the user's full name into the corporation user's object.

The **Role** property is a dropdown list of all Roles available in your App. It is recommended that you create a specific Role for the user on your **User Preferences Provider** App (e.g., User). Defining a Role will allow you to set Role level permissions for the user and help organize Phases in your user approval workflow.

Sending an Onboarding Email to New Users

Users that will not log in via SSO must authenticate via an email address and a password. Whenever a new Record is created in the **User Profiles** app for a brand new non-SSO user, you should configure a Conditional Compound Action (CCA) to generate a password reset token, and email the user, welcoming them to Onit. This email should also contain a link that allows the user to set up their Onit password.

In this situation, technically the user must *reset* their password in order to log in for the first time. Onit provides an Action that generates a password reset token named **Create Reset Password Token**. Once a token is created, via the Action, a Liquid filter named `reset_password_url` becomes available. Filtering the user's email with `reset_password_url` will generate the user's password reset URL. Therefore, it is necessary to fire the **Create Reset Password Token** Action *before* the **Send Notification** Action.

Below are examples of the **Create Reset Password Token** Action, **Send Notification** Action, and the **CCA** which fires both Actions:

Create Reset Password Token

Name *

Reset Password Token

Description

User Email

{{name}}

Editor

Comment

Provide a brief description of the change you made (optional)

Save

Send Notification

Name *

Send New User Notification

Description

Test Mode

☐

Test Email

From Name

{{corporation.name}}

Editor

From Address

{{email_address.account}}@{{email_address.subdomain}}.app.onit.com

Editor

Reply-To Name

{{corporation.name}}

Editor

Reply-To Address

{{email_address.account}}@{{email_address.subdomain}}.app.onit.com

Editor

Recipient Roles

Restrict Roles to Phase

☐

Recipient Emails

Recipient Email Fields

User Email (name)

Layout

Standard

Subject *

Welcome to Onit, {{user_name}}!

Editor

Body *






<p>Hello {{user_name}}</p><p>Welcome to Onit! Please use the link below to set your Onit password. </p>{%assign url = {{name}} | reset_password_url %}{{'Set your Onit password' | button_link: url }}</div></div></div></div>

History Name

Automating User Onboarding

Page 7

Conditional Compound Actions

Name *	Reset New User Password Email CCA	
Description		
Condition		 +
Actions *	<div>Reset Password Token </div> <div>Send New User Notification </div>	 +
Run In Background	<input type="checkbox"/>	
Action to execute in case of background failure		
Comment	Provide a brief description of the change you made (optional)	

Save

Notice the Liquid in the **Send Notification** Action,

```
{% assign url = {{name}} | reset_password_url %}{{ 'Set your Onit password' | button_link: url }}
```

This Liquid uses the generated password reset token from the Action and the `reset_password_url` filter to create a password reset link.



Make sure that "Run In Background" is *not* checked in the CCA. These actions will not work correctly when run in the background.

Automatically Assigning a User to a Suite

Once a new user Record has been created in the **User Profiles** App, the related user can be automatically assigned to a Suite by associating them to a **Group Preferences** Record. This can be especially helpful if you want all automatically created users to default to a specific Suite.



If you are unfamiliar with how to assign users and groups to Suites, [see our tutorial on this topic](#). The configuration instructions below will only makes sense if you're familiar with the concepts in that tutorial.

To programmatically assign a user to a Suite, use a **Find or Create Related Transaction** Action in the **User Profiles** app. This Action should relate the **User Profiles** Record to the appropriate **Group Preferences** record, by

way of setting the **group_preferences** Field. The example configuration shown below is configured to relate new (*auto-created*) users to the **HTIU Group**. Pay particular attention to the **Params Liquid** property. The `r_users_id:{{id}}` Liquid expression relates the user's Record via ID (`{{id}}`) to the appropriate **Group Preferences** Record (`r_users_id`).

Find or Create Related Transaction

Name *	Relate to Group Pref Record
Description	
Related Property Name *	Group Preferences (group_preferences)
Operation *	Upsert
Relation Field Name *	name
Relation Field Value *	{% if group_preferences == blank %}HTIU Users{% else %}{{group_preferences.name}}{% endif %}
Params Liquid *	r_users_id:{{id}}
Comment	Provide a brief description of the change you made (optional)

CopyDeleteSave

As seen above, it's important that the **Relation Field Value** property contain an if condition, not simply a hardcoded group name. The reason why is somewhat complicated, and it is related to the fact some **User Profiles** Records are created automatically while others are created manually.

Let's consider two examples:

- **Auto Creation:** A user named Mike is automatically created in Onit when Mike logs in via SSO for the first time. As a result, Mike's corporation user object is automatically created, which in turn creates Mike's **User Profiles** record, which in turn triggers the **Find or Create Related Transaction** Action shown above. The Liquid in the Action checks to see if the **User Profiles** Record is already related to a parent **Group Profiles** record, and since it is not it relates the child Record to the parent **Group Profiles** record named **HTIU Users**.
- **Manual Creation:** A user named Sally is manually created by an Onit administrator creating a new Record in the **User Profiles** app. When creating this Record, the administrator populates the App's Launch Page with various field values, such as the user's email address, full name, and their group. For the group, the user selects a value from the **group_preferences** Field's dropdown, which identifies which group they will belong to. In this situation, the **Find or Create Related Transaction** Action shown above will still fire, but we don't want to overwrite the group manually selected by the Onit administrator that created the Record – the if condition ensures this won't happen.

Out-of-office Delegate Feature

When a user leaves the office they can choose an out of office delegate to assume their Role(s) while they are gone. For more information on how to set this up, see our [out of office delegate tutorial](#).