

Onboarding for New Hires

1st Day - Important Links and Logins

Welcome to the team! First things first it's a good idea to get yourself setup with all the apps and websites that you'll be using on a regular basis. To make things easier we've collected some links to installers for apps and websites to bookmark.

Invitations from Manager

You will need to be invited to certain things in order to use them.

You should have email invitations to:

- Azure DevOps
- Revver (Production, Staging, Govcloud, UK, CA)
- Slack
- Discord
- Zoom
- Google Calendar (Recurring QA team / Product Development meetings and weekly 1-on-1's)

If you are missing any of these invites, please let Michael or another team member know!

Communication

This job involves a lot of communication, both within and outside our department, so get used to reaching out and asking questions. Our main app used for communication is Slack, however Discord is frequently used when communicating with the Dev team. Azure Dev Ops (ADO) is where the majority of our documentation for testing is located. It contains our Test Plans, Ticketing System, Work Items, Boards, and a multitude of other important information. It can be overwhelming at first but you'll get more comfortable with it as you use it.

Downloads

[Slack Installer](#) - To login you will use your work email address. Once you have slack installed you'll want to make sure that you're invited to the qa, qa-verification, and release-team channels in

addition to the general channels.

[Discord Installer](#) - You can login with either a personal account or create a new account with your work email address, but you will need to be invited to the eFileCabinet server to access it.

[Zoom](#) - You will need to set up a Zoom account with your work email.

[Teams](#) - You will log in with your work credentials (Microsoft, not Gmail) We no longer use Teams

[Remote Desktop Instructions](#) - You will need to remote into our Mac for testing as well as our other machines for running automated tests. If working remotely, you will need to talk to McKay in IT to get a VPN set up on your computer. Connecting from within the office will work automatically.

Websites to Bookmark

[Replicon](#) - Clocking in and out for Hourly employees and time off requests for Salaried employees.

[Azure Dev Ops](#) - Used for everything (bug ticket tracking, productivity boards and assignments, etc.)

[Gmail](#) - Company email. It is also useful to set up a second non-work email account for testing (ex. firstName.LastName.efc@gmail.com or efctesterLastName@gmail.com)

[Google Calendar](#) - Keep track of various meetings throughout the day. Can also be accessed through Gmail and integrated into Slack.

Rubex

There are four production Rubex environments, and one Staging environment where we do the majority of our testing. You should create bookmarks for all of these.

- [US Production \(Live\)](#) - Most of our customers are on this environment.
- [Staging](#) - Only Revver employees have accounts here. Staging will usually get multiple updates per week, while our Production environments only get updated a few times per month with builds that were tested and approved on Staging.
- [Govcloud](#) - An environment where some of our government customers have accounts. It has a slightly different security protocol than our other environments.
- [UK Production](#) - This environment is similar to our US Production environment, but the servers are based in the United Kingdom so our UK customers have better experiences speed-wise.
- [Canada Production](#) - Same deal as our UK environment, eh.

Google Sheets

Not everything we use lives in ADO, below are some Google sheets that we regularly reference.

[Rubex Post-Release Testing](#) — Used for tracking testing assignments for releases. Due to an increase in release cadence and the release Smartsheet, this has become obsolete.

[Rubex Staging Testing](#) - Used during swarms on Staging. Sometimes we'll do a "practice release" on Staging that mimics our release process onto our Production environments. This document is set up to match our Post-Release Testing document.

[Automation Progress](#) - Tracks our regression tests and which ones have been automated in Nightwatch and C#.

[Home/Office Assignments](#) - Spreadsheet that shows when to expect team members to be in the office or working remotely. **This is rarely used now.** Instead, when you work remotely, make sure to update your Slack status to show that you are working remotely for the day. Services like Zapier can be used to automate this, if you have a consistent WFH schedule.

[ADO Testing Discussion](#) — Notes to track changes to Test Plans while running through them. Sometimes you may encounter smaller spelling errors or be able to include clarification. Keep this open to make note of things while testing. This was mostly being used when we were developing our ADO test suite. We don't use this much anymore, but I'm leaving the entry just in case we start using it again.

Smartsheets

[Release Details Tracker Smartsheet](#) - Sheet that reports items in releases to other internal employees. Updated to match what we're releasing based on work items on staging.

[Last 30 Days Dashboard](#) - Easily digestible version of the Release Details sheet. Shows items in upcoming releases, last 30 days (duh.) and status of the top 10 QoE items.

[Customer Escalations \(Old QoE\)](#) - Sheet that shows items reported from the Customer Escalations team that they would be liked to be worked on based on customer complaints and friction.

[QoE Top 10](#) - Priority list of items from the Customer Escalations sheet. What is regularly discussed in weekly QoE meetings.

Other Useful Links

[Request Catcher](#) - Website for catching API Requests while testing

[QA Goodies](#) - Google Drive that contains:

- Folder of files to download for testing
- Videos for automation training
- Misc. documents

Once you have reviewed all of these resources and taken the appropriate related steps, go ahead and move on to the First Week section.

First Week - Getting a Taste of Revver and ADO

Goals

- Develop a basic understanding of Revver and how to navigate it
- Familiarize yourself with ADO Test Plans and run through a complete Test Suite
- Obtain a collection of files to use for testing

Files to Use

The most common file type for testing is a PDF, but it is important to have a collection of different types for regression testing. Check the QA Goodies link in the above Other Useful Links section for a wide assortment of files for use.

Getting to know Revver

Revver CSM Videos

[This link](#) will take you to a list of videos our CSM (Customer Success Management) team has put together to explain the basics and some advanced features of Revver. Don't worry about watching all of them, but treat it as a second reference if you have questions.

Rubex Exploration Through WalkMe

The best way to learn Rubex is through experience. Open Rubex in a new tab and login to your account. Once on the home screen, click on the orange "Need Help?" button in the top right. This will open a feature called WalkMe, which gives step by step tutorials for different Rubex functions. Take the time to complete each subsection listed in WalkMe and write down any questions that you may have as you do so. **Make sure you do this in our US Production environment**, not Staging, because the WalkMe content on Staging is experimental. Walkme is dead.

Revver Onboarding Test Plan

After finishing the tutorials in WalkMe and watching the tutorial videos, contact Michael or another team member and they will create a Test Plan for you to follow. This will give you more experience with ADO and further reiterate the things covered by WalkMe and the tutorial videos.

Test Plans is where you'll find instruction and guidance for completing the bulk of our work - testing!

Each Revver release needs to be thoroughly tested before it is released to the public, and our methods for testing are found in the Test Plans area on Rubex. We use a combination of testing types to provide the best coverage:

- Automated: Selenium tests that cover a wide variety of basic tests
- Smoke: Simple tests performed on other browsers to verify normal function
- Manual Regression: In-depth tests that cover a wide variety of factors and options

To access Test Plans, go to ADO and find the Test Plans section, or click on [this link](#). Once there, you should be able to view all Test Plans, both past and current. Let's use your onboarding test plan as an example.

With your test plan selected, you can access the different Test Suites within the plan. Selecting a folder (or suite) within a Test Plan will show any Test Cases under that section. If a folder shows the "Add a test case" message in the middle of the screen you may need to look in a sub folder of that section to find the tests.

With Test Case(s) visible, click "Execute" in the top center bar to switch modes. While in execute mode, click the checkbox next to the test cases, followed by the "Run for web application" button to run them. A new window will appear that will outline the steps to follow. Next to each step will be 2 circles - pass and fail. As you follow the steps of the test you may select these buttons to track your steps or if the test is simple you can click on the dropdown in the top right to select an outcome for the entire test. Within this window you can use the left and right arrow buttons on the top bar to navigate between tests that you selected, and once finished with the group of test you can click save and close.

That's the gist of it, but if you need any help don't hesitate to ask.

Some Advice:

- Avoid editing/deleting any tests in your test suite as they are linked to the template that we use for regression testing.
 - If a test says to "test for all item/file types", use your discretion to determine how many times you need to repeat the action.
 - Keep in mind that the goal of this first test suite is to give you practice and help you understand Rubex, not to rush to the end.
-

First Month - Becoming a True Member of the QA Team

Goals

- You are confident testing Revver on your own by following ADO Test Plans
- You are comfortable navigating ADO Boards, Work Items, Test Plans, and other commonly used areas
- You are comfortable writing bug tickets
- You are comfortable investigating internal questions (ex. qa-verification slack channel)

Testing

At this point you should feel comfortable navigating the different areas of Revver and completing basic tasks needed for testing. You may still have questions about some of the more complicated sections like Workflows, but that is completely normal. You should be able to easily follow Regression Testing plans similar to your Onboarding Test Plan but a little more in depth.

Using ADO

Since ADO is used as the central hub for our daily work it's important to understand how to navigate the different areas and know what they're used for.

Test Plans

When manually testing you'll spend a lot of time in the test plans section, and you'll likely use it as a reference when writing automated tests. Navigation here is fairly basic, you mainly need to know how to swap between different Test plans, how to switch between defining and executing tests, and how to create/add to bug tickets when finding bugs during testing.

Tickets

Speaking of bug tickets, let's talk about them. Whenever you find unexpected behavior within Revver it needs to be reported to our Dev team with as much detail as possible. The key things needed are:

- A descriptive title - You can't just say "Preview doesn't work right"
- Steps for reproduction - If you find the bug while following a test case you can create a ticket within the test window, and it'll even add in the test case steps for you. If doing it this way, make sure to mark which individual step(s) passes/fails and add any additional information needed in the comment window.
- Environment - If your bug is specific to a certain environment, ex. it only happens in CA, then make note of that in your ticket. Additionally use a tag such as [Search] or [Deep-Link] if the item is related to something a dev team is actively working on.
- System Info - Include the version number that you are testing (can be found by clicking the ? icon in the top right of Revver, then click About.) If the bug is specific to a browser, include which browser you were using. In addition, if the bug is specific to an operating system, include which OS you were using.

To access existing bug tickets, navigate to the Backlogs section under Boards on ADO and filter by Bug as the Work Item Type. **Search this area before reporting a bug just to make sure that it hasn't already been reported or fixed in a future build that hasn't been deployed yet.** Using the search functionality in the top right of ADO is a great tool as well, as long as you use filters to prevent code snippets from showing up in your search results.

Boards

Boards are where you will find what's currently being worked on by the team. If you're unfamiliar with the Agile/Scrum process things here may sound a little foreign, so here's some basic terminology.

- Work items - These can include tasks, bugs, features, or user stories. Basically items that need to be completed.
- Boards - A visual "wall" of sorts that tracks what's being worked on by the team. These tasks can be viewed at 3 different levels: Epics, Features, and User Stories. Epics tend to be more of a "end goal" that we want to accomplish. The steps we take to complete these epics are divided into different Features, which in turn are specified in User Stories. The purpose of this is to break down a large effort into tangible weekly goals.
- Backlogs - The area where all User Stories that are currently on our "To-Do" list live. If something needs to be done in the future, we create a User Story for it and it is placed here in the backlog. The backlog is sorted in order of priority, with urgent things placed at the top and less pressing (but still important) Stories placed at the bottom. This backlog is where we will pull in User Stories for our Sprints.

~~—Sprints— Sprints are simply periods of time where we work on specified Tasks that were decided on in Sprint Planning. Our Sprints currently run from Tuesday to Tuesday, with the goal of accomplishing all User Stories added to the sprint in this week long window. The sprint tab is where you can see what everyone is currently working on where you can find work that still needs~~

~~completed if you finish your tasks.~~ Our team has transitioned to more of a kanban-style system, so we no longer use sprints.

~~### Repos~~

~~This is where the Repository (AKA Repo) for Rubex lives. It is called Utopia, and all of our C# automated tests can be found in these files as well. You'll initially clone a version of Utopia for writing tests on your local machine from here, but later on the Repos section can be used to check different branches, pull requests, and other git related things. This has changed, now that Randy got the Selenium tests separated out from the rest of the Utopia code. For questions about our repo, talk to Randy Morris.~~

Overview

The first section listed is the last one I'll talk about. Recently we've implemented a Dashboard that gives an "at a glance" view of some important things. It's still a work in progress, so expect some more changes with it. The other important area in this section is the Wiki subsection. It contains a lot of helpful information on a variety of things, although some of it is likely outdated. Feel free to browse through these and make a mental note of which ones might be helpful in the future either for reference or troubleshooting.

Revision #4

Created 18 July 2022 22:33:23 by Bryce Holloway

Updated 21 March 2024 22:32:51 by Michael McGinnis