Designer User's Guide

Application workflow

2/19/2026

Contents

-

  - Administrator

Learn how to use builds and streams to manage the Designer application development workflow.

Related documentation:
  -

Overview

Designer uses a stream-based workflow for developing applications. In this type of model, an application consists of multiple work streams, each of which represents a different stage in the life cycle of the application:

- DEV (Development)
- QA (Quality Assurance)
- UAT (User Accessibility Testing)
- LIVE (Live Production)

When you want to freeze the current state of the application code, you can generate an application build and assign it to a stream. The build can then be promoted to the QA and UAT streams until you are ready to assign the build to LIVE and run it in the production environment.

This type of workflow enables you to have multiple builds of the application running in different streams, completely independent of the others. For example, a developer can work on a build without impacting the build that is assigned to quality assurance or the build that is actively running in production.

This video shows the workflow in action:

Link to video

Application streams

You can view and manage the streams of an application from the application properties:

There are four application streams:

DEV (Development)

This stream always runs the latest published version of the application code. As such, it is the only stream affected by Designer upgrades.

QA (Quality Assurance)

For builds that require Quality Assurance testing.

UAT (User Acceptance Testing)

For builds that require beta-testing or some other variation of user-based trials.

LIVE

For builds that are actively running in the production environment.

LIVE B

This is an optional stream that can run a second application build in the live production environment.

For more information, see LIVE B streams.

All streams are enabled by default. You can use the **Status** sliders to enable or disable them as needed.

Application builds

A build is basically a self-contained package of the application code. It has all of the resources it needs to run, so you can assign a build to a stream without worrying about impacts to the original resources or to the other builds.

Builds can be created only *after* the application has been published. If your application has not yet been published, the **Build** button remains disabled until the application has been published at least once. When generating a new build, Designer always uses the most recently *published* version of the application code. If you've made changes to an application that you want to capture in a new build, you must re-publish the application before generating the build.

To create a build, open the application in editing mode and click **Build:**

Designer automatically adds or increments the **version** of the build. You must assign the build a **Label** and, if desired, you can use the **Note** field to provide any additional information about the build. When you are finished, click **Build**.

Things to keep in mind when working with builds:

- **Builds are generated from the latest *published* version of the application code.** If you've made changes to an application that you want to capture in a new build, you must re-publish the application before generating the build.

- **Builds are immutable.** Once generated, a build cannot be changed.

- **Rollbacks are permitted**. To perform a rollback, simply assign a previous build to a stream.

- **Each application has a 20-build limit**. Therefore, it's recommended to only generate builds when required. If you exceed this limit, the **Build** button becomes disabled. You can re-enable the button by deleting builds that are no longer needed.

Important
Application builds must ideally have consistent logic and flow in all application

> streams. In other words, application logic should not be sensitive to the stream it is running on. An application build that exhibits problems in a Live stream will typically be loaded on non-Live streams for troubleshooting. If it changes behavior in the non-Live stream, it may not take the same path it does on the Live stream, which can complicate troubleshooting. It also implies Live stream behavior cannot be tested until the application is made Live, which goes against the streams based workflow that is designed to reduce such risks. It is normal for applications to access different sets of resources in Live and non-Live streams and that aspect is supported by the Parallel Test Environment (PTE) feature.

To manage the builds for an application, use the Manage Builds option when viewing the application properties.

Parallel Test Environment (PTE)

The Parallel Test Environment enables you to use test resources instead of production ones when running an application build in a non-production stream.

To enable this option, go to the Misc tab in the application settings and select **Enable** for PTE. You can then create copies of certain resources and add a special *test_* prefix to the resource name.

Streams that are running non-production builds (DEV, QA, and UAT) will reference the PTE versions of the resources (test_) instead of the ones being used in production. The LIVE build continues to reference the original production resources.

PTE is supported for Business Controls resources (**Business Hours**, **Emergency Flags**, **Special Days**, and **Data Tables**) and certain configuration resources (**Virtual Queues**, **Agents**, **Agent Groups**, and **Skills**).

LIVE B streams

You can run an optional LIVE B stream in addition to the existing LIVE stream. This allows you to run a second application in LIVE production mode, which gives you greater flexibility in how you can introduce new application builds into your production environment.

For example, each LIVE stream can have different contact points assigned to it, so one way you could use the LIVE and LIVE B streams is to allocate interactions coming from a certain region to a particular stream, or use the streams to balance (or gradually introduce) the number of interactions being handled by a new production build. In this way, you can implement a form of A/B testing.

Sample scenario

Let's say you wanted to split or balance the number of interactions being handled between the LIVE and LIVE B builds. Typically, you would gradually introduce more interactions from LIVE to LIVE B, which could be done as follows:

1. Assign **Build X** to the LIVE stream.

2. Use the **Manage** button to assign phone numbers to the LIVE stream. At this point, 100% of the interaction load will be handled by **Build X** on the LIVE stream.

3. Next, assign **Build X+1** to the LIVE B stream.

4. Assign some of the phone numbers to the LIVE B stream so that 70% of the interaction load is handled by LIVE and 30% by LIVE B.

> **Tip**
>
> The Transfer from Others button is a great way to easily and quickly move several numbers from one stream to another.

5. Check Designer Analytics to see if there are any issues. If there are no issues, we can move more of the phone numbers from LIVE to LIVE B so that 30% of the interaction load is being handled by LIVE and 70% load by LIVE B.

6. Continue to monitor Designer Analytics for any issues. If everything still looks ok, we can move the remaining phone numbers from LIVE to LIVE B so that 100% of the interaction load is being handled by LIVE B.

7. Assign **Build X+2** to the LIVE stream. This is a new build that we want to test in conjunction with **Build X+1** on LIVE B.

8. Assign some phone numbers to the LIVE stream so that 30% of the interaction load is being handled by our new application build running on LIVE and 70% by LIVE B.

9. Continue to check Designer Analytics for any issues, and adjust and balance the interaction load between the two builds as desired.