

# Using Build Administration

DTP enables you to compare changes in the codebase as it evolves. If your organization builds software continuously, then using builds for comparison provides insight that wouldn't be possible using a traditional date-based approach to comparing data. The Build Administration page provides visibility into build activity and information stored in DTP. In this section:

- [Accessing the Build Administration Page](#)
- [Locking and Archiving Builds](#)
- [Deleting a Build](#)
- [Build Audit Report](#)

Build Administration										
Filter: demo										
● Test, coverage or metric details are available. ○ Test, coverage or metric details are no longer available.										
Search										
Build	First Run Date	Runs	Static Analy...	Metrics	Tests	Coverage				
demo-2019-01-23	2019-01-23 15:00:38	1	1	0	0	0				
demo-2018-04-26	2018-04-26 11:44:00	1	1	0	0	0				
build-2	2018-01-28 02:30:00	3	3	0	0	0				
demo-new	2017-03-04 11:27:52	2	0	0	2 ●	0				
build-2015-07-28 11:15:42	2015-07-28 11:15:42	2	0	0	0	1 ●				
build-2015-07-23 12:10:54	2015-07-23 12:10:54	1	0	0	0	0				
build-2015-07-23 09:30:49	2015-07-23 09:30:49	1	1	0	0	0				

## Accessing the Build Administration Page

Click on the [Build Administration - Statistics](#) widget to open the build administration page.

Build Administration	
Total Builds	1156
# Archived	46
# Locked	16
Last 30 days	21

If you know the filter ID, you can manually enter the report URL:

`http://<host>:<port>/grs/dtp/reports/builds?filterId=<id>`







## Locking and Archiving Builds

You can lock a build to prevent DTP from accepting new reports tagged with the same build ID. Locked builds commonly mark the end of a development phase. You can archive a build to save the XML report files associated with a build ID, including the test and coverage details. DTP only keeps test and coverage details for the last two builds by default, but you may want to archive all data associated with a specific build so that existing test and coverage details are not deleted.

**i** **Archive a build to download reports**







A build must be archived in order to download the XML, HTML, or PDF version of the report. See [Build Audit Report](#) for additional information.

1. Add the Build Admin widget to your dashboard and click on a segment you want to investigate to open the Build Administration page. Make sure that the widget settings are configured to the appropriate filter. See [Build Administration - Statistics](#) and [Adding Widgets](#) for additional information.
2. Click one of the action buttons to lock, archive, or delete a build (also see [Deleting a Build](#)).

Metrics	Tests	Coverage			
0	0	0			
0	0	0			

All XML report files associated with archived builds are preserved in the `[DTP_INSTALL]/data/_DEFAULT_/dc/baselined/{build primary id}/{run id}` directory, where `{build primary id}` is the ID (primary key) of the build in the database and `{run id}` is the ID (primary key) of the run in the database.

You can unlock or release a build by clicking the updated icon.

Metrics	Tests	Coverage			
0	0	0			
0	0	0			

When the build is released, the XML report files will remain in the subdirectory until DTP Data Collector prunes the system. Data Collector checks for data that can be pruned several times a day.

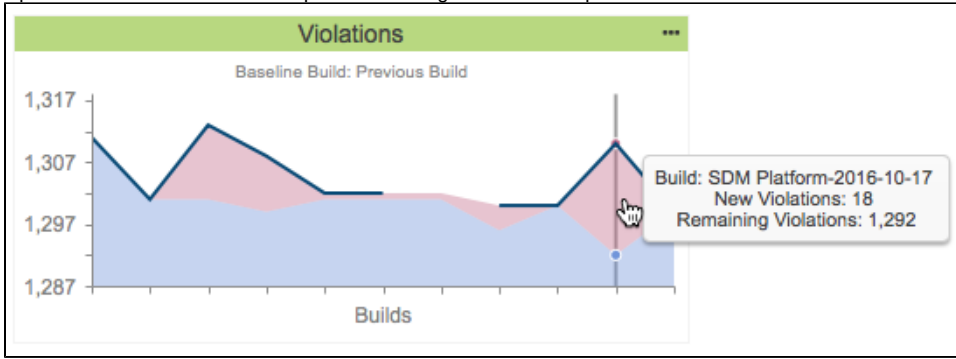
## Deleting a Build

If the data associated with some runs becomes unreliable, you can search for builds associated with the compromised runs and use the REST API to delete the data from the database. For example, if the source control system goes offline, then authorship, branch information, and other data may lose integrity and have a negative impact on the effectiveness of reporting mechanisms in DTP. If the system under test (SUT) experiences a critical failure, the test results associated with runs that were executed during the failure may represent "noise" and not actual regressions.

Data removed from the database is at the build level, so all data associated with the build will be deleted—even if the removal occurs across different runs. By default, Parasoft code analysis and test execution tools generate unique builds daily, so deleting data associated with a build will likely result in deleting all data for that day.

1. **Find the date of the data that should be deleted:** If you already know which build(s) you want to delete, [skip to the next step](#). This may be the case if you have the XML report associated with the data you want to delete. You can find the build in the root element of the XML report. There

are several ways to find the date of the data you want to delete. If your dashboard contains a trend widget, you may notice an unusual spike or dip in the results. Mouse over the point in the widget and the tooltip will show the exact date.



2. **Delete the data using the Build Administration page:**

- a. Add the Build Admin widget to your dashboard and click on a segment you want to investigate to open the Build Administration page. Make sure that the widget settings are configured to the appropriate filter. See [Build Administration - Statistics](#) and [Adding Widgets](#) for additional information.
- b. Find the build associated with the date that should be deleted and click the trash icon button.

## Additional Details About Deleting builds

- You can only delete one build at a time.
- If Data Collector is processing reports while you are deleting a build, DTP performance related to deleting the build will be impacted. This is due to simultaneous database access.
- For optimal performance with large sets of data, turning Data Collector off prior to deleting builds is recommended, but remember to start Data Collector when finished.

## Build Audit Report

Click on a build in the Build Administration page to open the Compliance Report for that ID. The Compliance Report shows all artifacts associated with a baselined build, such as environment attributes, test configuration used, and test results, so that you can understand how well the build is tested. See [Build Audit Report](#) for details.