# CI Build History Enhancements

[**CI Build History Enhancements**](#_5svoctgbq4i) **1**

[**Background**](#_db7dhx221ivk) **1**

[**Requirements**](#_1grzsnumwk4g) **2**

[**Current Status**](#_llg3psjqgigo) **2**

[**Proposal**](#_zanut5bsr3c) **3**

[Configuration Requirements](#_97mkac9dt3oq) 3

[Configuration Location](#_vibyl5qauh8y) 3

# Background

The decent\_ci build system requires a separate github repository (generally called something like project\_name\_results) where results from builds are stored. The results repository is used as a synchronization mechanism to ensure that multiple “build bots” are not duplicating work. The general process is as follows:

1. Identify a potential build that does not currently have a results stored
2. Attempt to create a “pending” results file in the results repository, which uniquely identifies the build to perform, taking into account branch name, architecture, settings, and compiler
3. If the attempt to create the pending file succeeds, perform build and post final results
4. If the attempt to create the pending file fails, skip and continue to the next potential build

The accumulation of result files in the results repository can lead to github failing to respond to api requests after the repository gets “too big.” This can lead to ci builds that are unable to complete because they cannot post or verify existing results files.

The definition of “too big” is unknown and seems to be dependant on limitations set (and not published) by github.

# Requirements

**CI Builds History Enhancements**. Currently, the CI Framework maintains a history of the builds being generated. The Subcontractor shall enhance the history functionality of the CI framework to include splitting the number of builds to keep in the history, based on the GitHub branch. The history for the default branch “develop” shall keep a longer history than feature and bug fixes branches, to allow investigation of problems that may have occurred in the past, and therefore being tracked through history spanning a longer period of time. The history length settings shall be easily modifiable and documented, to allow any NREL EnergyPlus development team member to independently perform changes to such settings. The Subcontractor shall provide a design document with proposed improvements, including proposed default lengths of history for different types of branches, and implementation details. This document shall be delivered to the NREL TM and distributed to the EnergyPlus development team for a peer review period. The Subcontractor shall address comments from this peer review period in the completed implementation of the CI build history improvement task.

The effective goal is to minimize the risk of having a large results repository make the decent\_ci build system become unresponsive.

# Current Status

There is currently a mix of hard coded and configurable limits within the decent\_ci system designed to mitigate the risks of the results repository getting “too big.”

The results limits are calculated on a per-branch (folder in the results repository) status.

If the folder contains more than 800 files:

* Each branch history is limited to 5 files per configuration
* Branch history files are limited to 60 days old

If the folder contains more than 999 files:

* Each branch history is limited to 1 file per configuration
* Branch history files are limited to “maximum\_branch\_age” + 1 day old

These rules help to limit each individual folder from getting past the known maximum of 1000 entries, but does not address the overall size of the results repository.

# Proposal

To address the overall health of the results repositories, the entire size of the repository must be taken into account. That is, the fail-safe numbers that are mentioned in the “Current Status” section must be adjusted to address the total file count in the results repository, not just the files in a given directory.

* If a configurable total file limit is reached (default value: 5000):
  + Feature/bug fix branches are limited to a configurable number of results per configuration (default: 5)
  + Long running branches (default: master, develop) limited to a separate configurable number of results per configuration (default: 20)
* If an individual folder hits one of the fail-safe limits, the hard coded fail safe mechanisms still apply
  + If the folder contains more than 800 files
    - Each branch history is limited to 5 files per configuration
    - Branch history files are limited to 60 days old
  + If the folder contains more than 999 files:
    - Each branch history is limited to 1 file per configuration
    - Branch history files are limited to “maximum\_branch\_age” + 1 day old

## Configuration Requirements

* Total file limit: (default 5000)
* “Long running branches names” (default: master, develop)
* Feature / bug fix branch files per configuration limit: default 5
* Long running branch files per configuration limit: default 20

## Configuration Location

The `security\_enhancements` branch to decent\_ci adds the notion of a “trusted branch” which can be used to pull trusted information from. The trusted branch defaults to the github default branch (currently `develop` for EnergyPlus).

In the case of the `security\_enhancements` work, the file `.decent\_ci-external\_users.yaml` is loaded to determine which users can be trusted to have CI built and run.

The recommendation is that we continue this model and have a new file called `.decent\_ci-limits.yaml` that is loaded from the trusted branch.