

# GEOS-Chem Release Proposal

Lizzie Lundgren, Melissa Sulprizio, and Bob Yantosca

GCST / Software Eng WG Co-Chairs

June 2021

## Overview

GEOS-Chem releases tend to drag on which is frustrating for everyone: GCSC, GCST, and users. We propose that we release GEOS-Chem more frequently and with a fixed schedule. Features would be rolled in continuously as they are ready rather than bundled in advance per version as they are now. Updates may not go into the model for one or more versions after delivery to GCST but the version release schedule would be reliable and consistent.

This document outlines a proposal for how we can accomplish this. It is presented in three parts: (1) restructure model priority tables to allow a more flexible development pipeline, (2) draft a new GEOS-Chem version release timeline and procedure that is realistic for GCST and satisfactory to all, and (3) develop a strategy for reducing benchmark time.

## Restructure [Model Development Priorities](#) tables

### Proposed New Tables

1. “Merged and awaiting release”
  - Features that are fully done (merged and tested by GCST)
  - All features will be included in next X.Y release
  - Replaces tables with specific version numbers
2. “Feature integration in progress”
  - Features being worked on by GCST
  - Work may include code development, pull request merge, merge conflict resolution, run directory updates, integration testing, difference testing, etc.
  - Estimated time to completion may vary a lot (e.g. less than 1 day to months)
  - Order of work determined in consultation with Model Scientists with consideration of GCST judgment, benchmarking needs and Working Group priorities
3. “Feature integration in the queue”
  - Features awaiting GCST work and identified as a priority
  - Inclusion determined in consultation with Model Scientists
4. “Feature integration not yet prioritized”
  - Features awaiting GCST work but not yet identified as a priority

## Comparison of New to Current

Current Table	Current Columns	Proposed Table	Proposed Columns
13.2.0	<ul style="list-style-type: none"> <li>• Feature</li> <li>• Submitted by</li> <li>• Type</li> <li>• Notes</li> </ul>	Merged and awaiting release	<ul style="list-style-type: none"> <li>• Feature</li> <li>• Contributor(s)</li> <li>• Model scope</li> <li>• Notes</li> </ul>
13.3.0			
Ready to go in		Feature integration in progress	<ul style="list-style-type: none"> <li>• Feature</li> <li>• Contributor(s)</li> <li>• Model Scope</li> <li>• Ease of implementation</li> <li>• Notes</li> </ul>
		Feature integration in the queue	<ul style="list-style-type: none"> <li>• Feature</li> <li>• Contributor(s)</li> <li>• Model Scope</li> <li>• Ease of implementation</li> <li>• Notes</li> </ul>
		Feature integration not yet prioritized	<ul style="list-style-type: none"> <li>• Feature</li> <li>• Contributor(s)</li> <li>• Model Scope</li> <li>• Ease of implementation</li> <li>• Notes</li> </ul>
Almost there (<6 months)	<ul style="list-style-type: none"> <li>• Item</li> <li>• Author(s)</li> <li>• Type</li> <li>• Notes</li> </ul>	Almost there (<6 months)	<ul style="list-style-type: none"> <li>• Feature</li> <li>• Contributor(s)</li> <li>• Model Scope</li> <li>• Notes</li> </ul>
Over the horizon (6-12 months)	<ul style="list-style-type: none"> <li>• Item</li> <li>• Author(s)</li> <li>• Type</li> <li>• Notes</li> </ul>	Over the horizon (6-12 months)	<ul style="list-style-type: none"> <li>• Feature</li> <li>• Contributor(s)</li> <li>• Model Scope</li> <li>• Notes</li> </ul>
Longer term (12-24 months)	<ul style="list-style-type: none"> <li>• Item</li> <li>• Author(s)</li> <li>• Type</li> <li>• Notes</li> </ul>	Longer term (12-24 months)	<ul style="list-style-type: none"> <li>• Feature</li> <li>• Contributor(s)</li> <li>• Model Scope</li> <li>• Notes</li> </ul>

## New Table Column Descriptions

- “Feature”
  - brief descriptions with link to GitHub feature request
  - Features should include both science and noteworthy structural updates that originate within GCST
- “Contributor(s)”
  - names and institutions
- “Model Scope”

- Type: science, structural (no diff), or bug fix.
- Framework(s): e.g. GC-Classic, GCHP, or CESM
- Sub-bullets with brief description of components impacted, e.g. mercury diagnostics
- “Ease of implementation”
  - Estimated level of GCST work to integrate feature: easy, moderate, or difficult
- “Notes”
  - Special notes provided by contributor
  - Information GCST feels should be communicated
  - Benchmarks required

## New Version Release Timeline and Procedure

### X.Y.Z

- Release as needed for important bug fixes
- Unlimited number of Z versions per X.Y
- Criteria for inclusion in Z version
  - Fixes for bugs that do not change the full chemistry benchmark simulation output
- Procedure:
  - Bug fix determined to be needed immediately
  - Create branch off of main
  - Add bug fix
  - Run all tests (CI, integration, difference) and perform other validation as needed
  - Create release page on wiki
  - Tag version
  - Push to main and send release notification

### X.Y.0

- Target release every 3 months
- Planned release dates are three weeks prior to GCSC quarterly meetings
- Criteria for inclusion in Y version
  - Tested and merged into dev prior to code freeze TBD weeks before release date
  - Features not yet merged should not hold up benchmark and release
- Procedure:
  - Ongoing:
    - GCST will work on feature integration on a rolling basis
    - Order of work determined in consultation with Model Scientists with consideration of GCST judgment, benchmarking needs and Working Group priorities
    - Integration testing and internal 1-month benchmark (private) done for each merged feature. Tag with X.Y.0-alpha.N.
  - 1-month before target release date:

- Assess “Feature integration in progress” table. Can any be wrapped up?
  - Assess “Merged and awaiting release” tables: What benchmarks are required?
  - Set code freeze date for dev branch based on benchmarks needed.
  - Create version page on wiki
- On code freeze date:
  - Freeze dev branch (no additional merging allowed)
  - Run all tests
  - Push fixes as needed based on test results
  - Update version page as needed
  - Tag version and start benchmarks
- When benchmarks complete:
  - Update version page with benchmark results
  - Share results with GCSC
- Next action predicated on GCSC comments on benchmark:
  - If no problems
    - Release on target release date or earlier
  - If problems
    - If problem source is known
      - If fix immediately available
        - If impacts only 1-month benchmark
          - Fix, tag, and rerun
        - If impacts other benchmarks
          - Handle on case-by-case basis
      - If fix is not immediately available
        - Handle on case-by-case basis
    - If problem source is unknown
      - Handle on case-by-case basis

## Reduce Benchmark Time

### Diagnostic Time

Diagnostics currently take up about 25% of run-time for full chemistry benchmarks. This can be reduced by limiting diagnostic output to only data needed for benchmarking. We propose changes to the following diagnostic collections in benchmark runs:

- Budget
  - Disable for both 1-year and 1-month benchmarks
- DryDep, ProdLoss, WetLossConv, and WetLossLS
  - Disable for 1-month benchmarks
  - Enable for 1-year benchmarks, but only output Ox species
- CloudConvFlux, ConcAfterChem, KppDiags, LevelEdgeDiags, StateChm

- Disable for both 1-year and 1-month benchmarks
- StateMet
  - Limit to only output fields needed for post-processing

## Spin-up Time

We plan on reducing spin-up time for the 1-year benchmarks from one year to six months. This will reduce the 1-year benchmark run-time by 25%. The initial July restart file will be taken from the previous 1-year benchmark.