National Centers for
Environmental Prediction

NCEP Central Operations

Software Version Numbering v3.0.0

Disclaimers

The information contained in this document is subject to change without notice.

Who Should Read This Document?

This guide is designed to benefit the following professionals:

System/Subsystem Owners and Team Members

This document will help the system owners apply the rules for version numbering consistently across all operational software within NCEP Central Operations

Configuration Managers

This document will help the configuration managers enforce the rules for version numbering consistently across all operational software.

Document History

Paper copies are valid only on the day they are printed. Contact the author if you are in any doubt about the accuracy of this document.

Revision History

Revision Number	Revision Date	Summary of Changes	Author
1	5/10/2010	Initial Document	Scott Jacobs
2	6/29/2010	Add Software Units and Initial Version Numbers	Scott Jacobs
3	7/2/2010	Update Mesoscale Branch information	Geoff DiMego

4	7/2/2010	Update Global Branch information	John Ward
5	7/6/2010	Update NCO library information	Scott Jacobs
6	7/7/2010	Update Marine Branch information	Hendrik Tolman
7	7/7/2010	Re-order sections	Scott Jacobs
8	7/28/2010	Sponsor corrections and suggestions added	Scott Jacobs
9	7/29/2010	Added “Major” Systems to the software unit tables	Scott Jacobs
10	8/18/2017	Added “pre-release” version number Make generic to NCO instead of WCOSS specific Removed section 3&4	Steven Earle
11	9/16/2020	Updated Section 1.2 Re-wrote Sections 2.2, 2.3, 2.4 Removed Sections 2.5 and 2.6	Joshua Huber

Approvals

This document requires following approvals:

Name	Title
Ben Kyger	Director, NCEP Central Operations

Table of Contents

1. Introduction

1.1 Purpose of this document

This document covers the scheme used to number the software units used in production across NCEP Central Operations. This document will also identify the software units and define the boundaries of each unit.

1.2 Identification

- This document applies to all software units currently used in NCEP Central Operations (NCO) and to any future units developed or implemented by NCO.
- The numbering scheme closely follows the Semantic Versioning standard version 2.0.0.

2. Version Numbering Scheme

2.1 Overall Scheme

There are a wide variety of numbering schemes used in software development projects. They are all variations on the same basic concept: keep track of different versions of a piece of software. The vast majority of the version numbering schemes are sequence based. A sequence of numbers, separated by a delimiter, is used to convey the significance of the changes between releases. The left-most value is changed for the most significant software modifications. Changes to the subsequent values indicate decreasing significance.

For applications maintained by NCO, all production software will take on the following form:

```
major.minor.patch
```

Where major, minor, and patch values are integers.

For planning purposes, the next version number of an application should be defined early in the development process. This will help in the messaging to stakeholders as to what to expect next from the application. If a new requirement is added during development or a requirement is removed from the scope, the version numbering should be reviewed to ensure it aligns with the major, minor, patch paradigm.

With the delivery of an application package, this becomes the baseline of the next release. Any pre-release changes to the baseline will require a new version number to be applied per this standard.

Each of the following sections defines the criteria for increasing the value of each sequence number.

2.2 Major Version

A major version is simply a breaking change to the application functionality. This generally means the user must change the way they are requesting or handling data responses. Major version examples include:

- Model resolution changes.
- Underlying model replacement and/or changes.
- Major scientific technology upgrade.
- Major version update to an international standard.
- Table format changes.
- Removal of current data locations or request points.
- Moving data locations if not first deprecated.
- Automatic parsing of the data being impacted.
- Additions to the data that alter how it could be potentially parsed or manipulated.

2.3 Minor Version

A minor version is something that is backwards compatible with existing functionality. Generally this will mean enhancing the data return by adding fields or fixing bugs related to existing fields in data response. From a user perspective this should be completely transparent. Minor version examples include:

- Model Physics changes.
- Model execution changes.
- Table entry updates.
- Removing deprecated data locations.
- Bug fixes or functionality additions that are transparent to the user.

2.4 Patch Version

A patch version has the least scope. This level of update is reserved for bug fixes or changes that are targeted to a single functionality and does not impact any other application function than the one under work. The patch number will usually be increased for solutions to operational problems that cannot wait for a minor or major version.

2.5 Summary

This standard is meant as a guide and cannot cover all possible scenarios. If you are unsure about where your code fits, please contact the appropriate team as listed below.

WCOSS - ncep.list.spa-helpdesk@noaa.gov

Onboarding - idp-support@noaa.gov

DataFlow - ncep.list.pmb-dataflow@noaa.gov