# Version number, naming conventions and file format of the RGI

## 0. Preamble

This document attempts to describe the file and version naming conventions of the RGI in the past (v6), what needs to change, and explain the rationale behind these choices.

**Note: it is not necessary to reach a decision quickly about ALL these changes. Most discussion points are largely independent of the others. If you are overwhelmed, focus on thinking about the RGI version number for now.  The two other "controversial" points are how to handle the different versions of the same RGI (merged/dissolved and default) and whether to allow for download of separate region files.**

## 1. Current format

This is the format of the RGI files as of version 6.0. Previous file formats are not discussed here since they will not change.

*Complete RGI zip file:*

```
00_rgi60.zip
```

*Complete RGI unpacked:*

```
00_rgi60_attribs/
00_rgi60_regions/
01_rgi60_Alaska/
(...)
19_rgi60_AntarcticSubantarctic/
00_rgi60_30-30grid.dat
00_rgi60_links
000_rgi60_manifest.txt
00_rgi60_summary.csv
00_rgi60_TechnicalNote.pdf
```

*Region files zip files:*

```
01_rgi60_Alaska.zip
```

```
(...)
19_rgi60_AntarcticSubantarctic.zip
```

*Region files unpacked:*

```
01_rgi60_Alaska.dbf
01_rgi60_Alaska.prj
01_rgi60_Alaska.shp
01_rgi60_Alaska.shx
01_rgi60_Alaska_hypso.csv
```

## 2. Issues with the current format

### 2.1 File names

Several reports from RGI users can be found on github:
https://github.com/GLIMS-RGI/rgi_issue_tracker/issues/9

It is quite technical but we can summarize the most important points here:

- Mixed use of CamelCase (e.g. TechnicalNote, GreenlandPeriphery) and snake_case (e.g. manifest) - often, consistency should be preferred - making everything in the same case avoids surprises.
- Combined use of rgi_code and a name that does not match the first-order region name (for example, Region 02 is called “Western Canada and USA” and the file is called “02_rgi60_WesternCanadaUS”, region 13 is called “Asia, Central” and the file is named “13_rgi60_CentralAsia”)
- Files that start with a number cannot be read properly by certain software tools.
- The current system is error prone (for example, the RGI6 Scandinavia folder used to be wrongly named “07_rgi60_Scandinavia” although it should be “08”. See this list for all known issues in the RGI file names themselves - now corrected)

Altogether, the main problems are that the files are not immediately machine readable (one has to curate a self-written list of filenames since they are not available in another list) and contain errors. **The process of naming files should be automated, like anything else in the RGI.**

## 2.2 Version names

The current format is to remove the point in the version name and use two digits, e.g. “3.2” becomes “rgi32” and “6.0” becomes “rgi60”. The problem with this format is that it won’t allow an RGI version 10.0 (this translates to 1.0) or a version 7.11.

# 3. New format of the RGI files

## 3.1 RGI version numbers (regardless of the file names)

This is a fundamental aspect of all RGI versions, and cannot be rolled back for reproducibility reasons. RGI has used Semantic Versioning with two digits since its beginning, and it will continue in the future[1].

**OPTION 1:**

1. **RGI v7.0 will be the final version targeting the year 2000. Future iterations of the year 2000 inventory will have the number 7.1, 7.2, 7.11, etc.**
2. **Future major versions of the RGI will target other reference years (e.g. v8 for 2015, v9 for 1850, etc.), with their own iterations.**

Advantages:

- This is transparent and consistent with previous RGI versions. Earlier versions were a clear improvement over the previous ones, RGI6 finally making authority. RGI7 is the last major iteration with a move to a transparent reproducible workflow.
- Minor increments (7.1, 7.2, etc.) will be understood as small updates to the same dataset while a new version (v8) will be a new dataset, i.e. targeting 2015
- Adding a new formalism for “RGI 2015” or “RGI 1850” will add confusion, especially because of the possible misinterpretation of a potential “RGI 2015” as the date of publication instead of the target year.

Inconvenients:

- New users might be confused by this (“why is RGI7.2 still valid despite of having RGI8.1”?). Ethan says: “better not to accumulate too many arbitrary ids with important meanings”
- The actual order in which the versions are released is not known (this happens in software as well). E.g. v8.0 can be released before 7.2.
- The target year needs to be known or read from a table

[1] Note that it would have been possible to use Calendar Versioning, which is a better choice in many aspects. But it's too late to switch now.

**<u>OPTION 2 based on comments by Michael, Romain and Ethan:</u>**

1. **Version 7 will never exist. It will be replaced by target years and incremental versions. For example: v7.0 will actually be called "RGI-TY2000.0", v7.1 will be "RGI-TY2000.1". Similarly, "RGI-TY2015.0" will exist, and maybe "RGI-TY1850.0".**

Advantages:

- The target year in the name is very explicit: much more than version 7 or 8.

Inconvenients:

- The actual "version number" gets a bit lost, i.e. compare "v7.1, v7.2" with "v2000.0, v2000.1". 2000.1 might be confused with decimal dates.
- It needs more characters.
- The actual order in which the versions are released is not known (this happens in software as well)
- It will take quite some time to get used to it, and it doesn't look like a version number (*"I used RGI v6, which version did you use? I used version 2000.0"*).
- It does not play well with the NSIDC system (but this we can arrange, i.e. the "_v7" url will point to the 2000s versions, and the "_v8" to the 2015 versions).

**<u>OPTION 3 based on comments by Regine:</u>**

1. **This is a mix of option 1 and 2. We add the target year AND the version number to the file. Therefore, RGI7 will become rgi_2000_v7.0 and the future RGI 2015 will become rgi_2015_v1.0**

Advantages:

- The best of both worlds listed above

Inconvenients:

- It gets a bit long.
- The question of incremental minor versions remains - do we need a v7.1 or should we do a version 8.0, in which case we could scrap the minor version digit altogether.

**This is now open for discussion.**

## 3.2 RGI version number in file names

This depends a bit on the decision taken on 3.1. Here are the options:

### A. Continue with the previous system (version number multiplied by 10 in the file name)

Advantages:

- Consistency with previous versions
- The files are (relatively) self explanatory, e.g. “02_rgi70_WesternCanadaUS” is version 7.0 (once one gets used to it)
- Important: this is consistent with the RGI ID system

Inconvenients:

- This cannot accommodate for a version 10 or a version 7.11
- This will be inconsistent with the system chosen at NSIDC, which has integer numbers in the header (daacdata.apps.nsidc.org/pub/DATASETS/nsidc0770_rgi_v1/)
- Inconsistent with the real version number (71 is not 7.1)
- Less readable than option B

### B. Use Semantic Versioning in the file name (example: rgi7.0_complete.zip)

Advantages:

- Very readable
- Consistent with the real version number
- It is more consistent with the system at NSIDC
- Can accommodate with version 10.0 and version 7.11

Inconvenients:

- It is a breaking change to previous conventions
- Important: might need a change in the RGI ID system
- If we ever go to version 10.0 or version 7.11, files in a folder won't be sorted properly (minor inconvenience)
- Maybe a problem for certain operating systems? (unlikely)

### C. Same as B but with a - (example: rgi7-0_complete.zip)

Advantages:

- Very conservative (no dot, no confusion with file ending)
- Same advantages as B

Inconvenients:

- Same invenvenients as B
- I don’t find it very pretty

D. **No mention version number at all in the file name**

This is a radical, yet preferred way for machines. The actual version number would be stored as metadata in the folder (e.g. a version.txt or similar to be discussed) and as attribute to the files.

Advantages:

- No headache about the file names
- Similar to software code, the name is "RGI" and the version matters less or can be queried if needed.
- Machines can read all RGI versions in the same way

Inconvenients:

- Humans won't find it that easy to infer the version of the dataset

**This is now open for discussion. I (Fabien) vote for B.**

## 3.3 RGI Region file names

Here again, several options:

A. **Keep the same names as before (but do not start with the region number, example: rgi7.0_02_WesternCanadaUS)**

Advantages:

- Similar to what was before, e.g. people won't be confused

Inconvenients:

- Inconsistent across versions (there were name changes in the past)
- The use of CamelCase may look nice but it is not consistent, and causes problems on Windows (two files named Western.txt and western.txt can't be in the same folder)

B. **Define a simple consistent system and follow it (example: rgi7.0_12_caucasus_middle_east.shp)**

The RGI would ship with a metadata file (already available in rgi70_O1Regions.shp files for example) which explains how the files should be named: a region would become a short_id (12) and a long_id (12_caucasus_middle_east).

Advantages:

- If consistent across versions, this would allow machines to read all RGI versions

- Predictable, human and machine readable
- `short_id` and `long_id` are common idioms and easily understood

Inconvenients:

- Maybe a bit too long filenames?

**C. Remove the region names from the files entirely (example: rgi6.0_12.shp)**

Advantages:

- Easy and machine readable

Inconvenients:

- Not human friendly (“expert mode”)

**This is now open for discussion. I (Fabien) vote for B.**

## 3.4 New RGI subversions: “merged” and “default”

With RGI7, we will have the default version of the RGI with all individual outlines, and a “merged” version with all outlines merged with dissolve where applicable. This will come with various challenges.

### In file names

The file names will need to reflect this. I suggest adding a flag to the file name for clarity:
rgi7.0_d_12_caucasus_middle_east.shp for “default”
rgi7.0_m_12_caucasus_middle_east.shp for “merged”

An alternative would be to not change the default files and add a flag for the “merged” ones:
rgi7.0_12_caucasus_middle_east.shp for “default”
rgi7.0_m_12_caucasus_middle_east.shp for “merged”

**This is now open for discussion.**
**FP: I strongly favour a)**

### In the RGI ID

It is fundamental to avoid confusion between the two IDs.

Currently we have RGI60-07.01543 for the “regular glaciers”. Pending the discussion above, this will become RGI7.0-07.01543 in the upcoming version. This will then need a new flavor for the “merged” version, which could be “RGI7.0-d-07.0154”

**This is now open for discussion.**

## 4. Discussion: rethink the RGI ID system

**There is absolutely no doubt that we will never change the existing RGI IDs (v6 and backwards).**

However, the discussions above illustrate that a revamp of the RGI ID system will be necessary starting with version 7. As long as the previous IDs don't change, I don't see a big problem in changing the new IDs. We will have to tackle this once the file names discussion is settled, but keep the IDs in mind when discussing the file names.

## 5. Discussion: disallow download of region files?

The "RGI complete" file is THE reference data. It contains the region files, metadata, the technical note, a copyright, and other attributes.

The RGI region files currently have only the hypsometry files and the shapefiles. They are missing all the metadata. So we have two options:

A. Making the region files *also* have the metadata. This would require some work to be done properly, but would be doable. These region folders will then be different from the ones you'll find in the "RGI complete" file - because they don't need it.
B. Disallow the download of region files.

This would be the best option in my view. Here are my arguments:

- with the new "merged" and "default" versions of RGI, the number of files to list will be (18+1) times 2. I would very much prefer to keep space for other options, for example to allow for download for "RGI without hypso", "RGI with subregion files", etc.
- region files are error prone because duplicated
- With the NSIDC move, all (18+1) times 2 files will get an "nsidc0770_" prefix
- There is only very little gain for the users. In 2022, it does not make much of a difference to download 413 MB once (size of RGI6). Users can organize their data how they wish after download.

**This is now open for discussion.**

## 6. Region names, and long ids

Table open for discussion here.