The OpenModelica Package Manager

Martin Sjölund

Department of Computer and Information Science
Linköping University

2022-01-31
Overview

Part I Current handling of libraries in OpenModelica
Part II New handling of libraries
Part III Scripting support
Part IV OMEdit support
Part V Conclusion
How packages are installed in current OpenModelica releases

▶ Windows
  ▶ A very large installer (1.4GB) containing all indexed libraries.
  ▶ Actually not true... There were too many for the installer, so only 1/8 of them are now included.
  ▶ Very slow to install the packages, so there is an option to only install some of the few now included.

▶ Ubuntu/Debian
  ▶ No library required by default.
  ▶ Option to apt install omlib-modelica-4.0.0 and some of the other indexed libraries.
  ▶ Confusing for new users.

▶ MacOS
  ▶ No longer well supported.
  ▶ Had the option of installing all libraries. Large download changing every day.
Which libraries were indexed?

▶ Many libraries tracked only the master branch and might break suddenly.
▶ Some libraries tracked older releases but were often outdated.
▶ Some libraries tracked multiple versions.
▶ When a new library was released, libraries depending on the older version needed to manually handle the uses.
▶ For Linux, libraries were updated automatically and might break on the installed release version with no ability to go back in versions.
Structure of the index

- https://github.com/OpenModelica/OMLibraries/blob/master/repos.json
- Libraries need to be packaged by OpenModelica, redistribution licenses checked, etc.
- Manually updating too many things.
- All libraries are then indexed based on this and packages created. Very slow.
Part II

New handling of libraries
Structure of the libraries to index

▶ https://github.com/OpenModelica/OMPackageManager/blob/master/repos.json
▶ Similar to the old index.
▶ Based on this, a JSON-file with cached information is generated. Updating that index is fast relative to the old system.
▶ Need to manually update the support level (new).
▶ No license information since OpenModelica will not store any library code on its own servers.
Structure of the libraries to index: Special options

- **ignore-tags** (sometimes the same version was released multiple times or does not contain any Modelica library)
- **semverTagOverridesAnnotation** (if tagged 3.2.3, assume the package is 3.2.3 even if the maintainer forgot to update the version since 3.2.0)
- **semverPrereleaseOverridesAnnotation** (if the library has a pre-release in the version, like beta.1 but is tagged with a different pre-release name like beta.2 - use the one in the tag)
- **git vs github vs URL to zips**
- **bitbucket-api-downloads-instead-of-tags** (like Github releases)
- **search-extra-paths** (if not stored at the root of the repository)
- **standard** (if a tag requires a different Modelica grammar to parse)
- **singleFileStructureCopyAllFiles** (passed to index.json; used for zip-files that do not store everything in a directory)
Structure of cached metadata

- https://github.com/OpenModelica/OMPackageManager/blob/master/rawdata.json
- Does not duplicate information from repos.json (support level) since entries are not scanned again later.
- Contains information about the uses and conversion annotations.
- A version number is given to each entry. Although this name is semver-style, the annotations are used to resolve which packages to install later.
- Old entries are not checked if they are still valid.
Structure of generated index

- https://libraries.openmodelica.org/index/v1/index.json (now 241 kB)
- Generated from repos.json and rawdata.json.
- Contains support-level (may update without scanning all entries again).
- Has a URL for a zip-file, hosted somewhere online (Github, self-hosted Gitlab, Bitbucket, or plain URLs).
- Has dependency information.
- Downloaded by updatePackageIndex() in OMC.

```json
{
    "Buildings": {
        "git": "https://github.com/lbl-srg/modelica-buildings.git",
        "versions": {
            "7.0.3-maint.7.0.x": {
                "convertFromVersion": ["6.0.0"],
                "path": "Buildings",
                "provides": ["7.0.0","7.0.1","7.0.2"],
                "sha": "b47602ee2a8d4d74440c18855dea29df0fb909c6",
                "support": "noSupport",
                "uses": {"Modelica": "3.2.3"},
                "version": "7.0.3-maint.7.0.x",
                "zipfile": "https://.../modelica-buildings/archive/b47.zip"
            },
            "8.0.0": {
                "convertFromVersion": ["7.0.0"],
                "path": "Buildings",
                "sha": "5fa0bed0caa0f27b8ebfa20c1a39e22cdd0f5008",
                "support": "support",
                "uses": {"Modelica": "3.2.3"},
                "version": "8.0.0",
                "zipfile": "https://.../modelica-buildings/archive/5fa.zip"
            }
        }
    }
}
```
Part III

Scripting support
Installing a package

updatePackageIndex();getErrorString();
installPackage(Buildings, "7.0.0");getErrorString();
true
"[/home/martin/.openmodelica/libraries/index.json:0:0-0:0:readonly] Notification: Downloaded package index from URL
  ➡  https://libraries.openmodelica.org/index/v1/index.json.
"
true
"[/home/martin/.openmodelica/libraries/ModelicaServices 4.0.0+maint.om/package.mo:0:0-0:0:readonly] Notification: Package installed successfully (SHA 144b3f853bdbaf0b7de5e127dbe0cfa738adf4f3).
  ➡  [/home/martin/.openmodelica/libraries/Complex 4.0.0+maint.om/package.mo:0:0-0:0:readonly] Notification: Package installed successfully (SHA 144b3f853bdbaf0b7de5e127dbe0cfa738adf4f3).
  ➡  [/home/martin/.openmodelica/libraries/Modelica 3.2.3+maint.om/package.mo:0:0-0:0:readonly] Notification: Package installed successfully (SHA 093b8cc9ae492e81f0766736ca6e413351319990).
  ➡  [/home/martin/.openmodelica/libraries/Buildings 7.0.2/package.mo:0:0-0:0:readonly] Notification: Package installed successfully (SHA 3d0f7d1a9cd14ba54985f024733b27b024774193).
"
How does OMC know to install ModelicaServices 4.0.0 instead of 3.2.3?
How does OMC know to install ModelicaServices 4.0.0 instead of 3.2.3?

```plaintext
package ModelicaServices
annotation (  
    version="4.0.0",  
    versionDate="2020-06-04",  
    dateModified = "2020-06-04 11:00:00Z",  
    revisionId="144b3f853 2022-01-01 00:39:47 +0000",  
    conversion(
        noneFromVersion="1.0",  
        noneFromVersion="1.1",  
        noneFromVersion="1.2",  
        noneFromVersion="3.2.1",  
        noneFromVersion="3.2.2",  
        noneFromVersion="3.2.3"),
end ModelicaServices;
```
Installing a package (exact match)

Not recommended, but can be used to force some known versions of libraries.

```plaintext
updatePackageIndex();
getErrorString();
installPackage(Buildings, "7.0.0", exactMatch=true);getErrorString();

true
"[/home/martin/.openmodelica/libraries/index.json:0:0-0:0:readonly] Notification: Downloaded package index from URL
  ↪  https://libraries.openmodelica.org/index/v1/index.json.
"

true
"[/home/martin/.openmodelica/libraries/ModelicaServices 3.2.3+maint.om/package.mo:0:0-0:0:readonly] Notification: Package installed successfully (SHA 093b8cc9ae492e81f0766736ca6e413351319990).
[/home/martin/.openmodelica/libraries/Complex 3.2.3+maint.om/package.mo:0:0-0:0:readonly] Notification: Package installed successfully (SHA 093b8cc9ae492e81f0766736ca6e413351319990).
[/home/martin/.openmodelica/libraries/Modelica 3.2.3+maint.om/package.mo:0:0-0:0:readonly] Notification: Package installed successfully (SHA 093b8cc9ae492e81f0766736ca6e413351319990).
[/home/martin/.openmodelica/libraries/Buildings 7.0.0/package.mo:0:0-0:0:readonly] Notification: Package installed successfully (SHA 903a06999fb514ebc1d0f58ef41b3f6dda0279ed).
"
```
Part IV

OMEdit support
Installing libraries (File⇒Install Library)

- Filtering based on release/post-release/pre-release.
- Filtering based on OpenModelica’s support level (of the latest release or nightly OpenModelica).
- Automatically installs dependent libraries.
- Exact match = do not install newer versions of libraries that state they are backwards-compatible.
- Ability to install older versions of libraries if necessary.
Part V

Conclusion
Future Work

- Better support for proxy servers?
- Dependencies on OpenModelica version (for Modelica/ModelicaServices?)
- Support for Github releases?
- Support for git-lfs? Only downloading some of the artifacts?
- Automatically compiling binary libraries on install?
- Adding better dependency resolution.
- Using the same dependency resolution for loading and installing libraries.
- Also consider conversion scripts.