

---

# MueLu Documentation

*Release 12.8*

**Andrey Prokopenko**

**Apr 25, 2017**



---

## Contents:

---

<b>1</b>	<b>New developer quick start guide</b>	<b>1</b>
1.1	Trilinos . . . . .	1
1.2	Using git . . . . .	1
1.3	Mailing lists . . . . .	1
1.4	CSRI . . . . .	1
1.5	MATLAB . . . . .	1
1.6	Extra . . . . .	2



---

## New developer quick start guide

---

Welcome on board!

### **Trilinos**

Checklist for new Trilinos team members.

### **Using git**

Can we create a new page here?

### **Mailing lists**

Developers Users Checkins

Other useful mailing lists: Xpetra, Tpetra, Epetra, Kokkos, Zoltan2, Amesos2, Teuchos, Framework, Users, Developers

### **CSRI**

CSRI wiki.

### **MATLAB**

Shared licenses of Matlab are available on both the SON and SRN. See CSRI wiki for details. Contact CSRI system admins to get access to MATLAB (nfs mount).

The point of contact for transferring personal MATLAB licenses is Ester Valenzuela.

## Extra

Paper/Presentation repository:

```
$ git clone software.sandia.gov:/git/muelupubs
```

## Debugging tools

### Debugging

Run GDB with input arguments:

```
`bash $ gdb --args ./MueLu_UnitTests.exe --test=MakeTentative `
```

Break on exception (in gdb): ` \$ catch throw `

Repeat commands in multiple instances of *gdb*: `bash \$ echo -e "catch throw\nrun" > /tmp/gdb-cmd \$ gdb -x /tmp/gdb-cmd --args ./MueLu\_ScalingTestParamList.exe [test args] `

### How to plot and debug factory dependencies

You need to have - *boost* (specifically *boost::graph* library) - *graphviz*

It is possible that you need to compile boost with shared libraries (*./b2 link=shared*).

You also need to update your Trilinos configure script: `cmake -D TPL\_ENABLE\_Boost=ON -D TPL\_Boost\_INCLUDE\_DIRS=<path to Boost headers> -D TPL\_Boost\_LIBRARIES=<path to Boost libraries> `

If you want the hierarchy to dump the dependency graph, you need to add this piece of code before the Setup function `c++ H->EnableGraphDumping("graph.dot", level); ` where *H* is your *Hierarchy* (*RCP* to it), “graph.dot” is the filename of the output file, and “level” is the dependency level (e.g., level 1 would dump dependencies for level 1 and 2).

The default value for level is 1.

After you have generated your output file, for instance, “graph.dot”, you need to process it. You can do that using the provided *utils/graph.sh*. That would create a “graph.ps” PostScript file.

how to use git