Trilinos v12 Deprecation Announcement

October 22, 2019

Announcement

In Trilinos v12.16, many features of Trilinos were deprecated. This document lists deprecated features, along with testing instructions and mitigation paths.

Expected schedule:

- November 15, 2019: Trilinos developers begin removal of code from Trilinos' develop branch
- December 4, 2019: Trilinos release v13

Concerns about these dates should be shared with Trilinos product owners and managers.

Deprecations

1. **Packages:** Trilinos-wide next generation stack
   Deprecation: Explicit instantiation of multiple different global ordinal types and multiple different local ordinal types
   Mitigation: Use default global ordinal type `long long` and local ordinal type `int`, or specify to CMake `-DTpetra_INST_INT_INT`
   to use global ordinal type `int` and local ordinal type `int`.
   Justification: Building with multiple ordinal types increases build times and library sizes; all applications surveyed used only one global ordinal type.

2. **Packages:** Tpetra
   Deprecation: `DynamicProfile` CrsGraph and CrsMatrix construction: new default construction type is `StaticProfile`
   Mitigation: Use `StaticProfile` construction, specifying (at least) the maximum number of nonzeros per row that the graph/matrix will hold.
   Justification: Memory allocations required for `DynamicProfile` are expensive on CPUs and infeasible on GPUs; supporting both static and dynamic construction makes Tpetra code base fragile.

3. **Packages:** Teuchos
   Deprecation: Use of `KokkosClassic::DefaultNode` and its synonyms `Kokkos::Compat::Kokkos*WrapperNode` as an object
   Notes: `KokkosClass::DefaultNode` and its synonyms may still be used as template parameters.
   Mitigation: `DefaultNode` objects are no longer used in Trilinos, so removing all constructed `DefaultNode` objects from application code should have no effect. Use `KokkosClassic::DefaultNode` typename (or its synonyms) only as a template parameter.
   Justification: `DefaultNode` is superceded by `Kokkos::ExecutionSpace` and `Kokkos::MemorySpace`.

4. **Packages:** Tpetra, Thyra, Xpetra, Ifpack2
   Deprecation: All functions taking a `KokkosClassic::DefaultNode` object as an argument
   Notes: In Trilinos, this deprecation most impacts `Tpetra::Map` constructors and their helpers (e.g., `Tpetra::createUniformCo`
   Mitigation: Call new replacement functions that do not have this argument. Function names and prototypes are identical with the exception of the `DefaultNode` argument's removal. Specification of the Node template parameter may be needed when calling the functions.
   Justification: `KokkosClassic::DefaultNode` is deprecated (see 3).

5. **Packages:** Tpetra, Xpetra, Ifpack2, KokkosClassic
   Deprecation: Methods `getDefaultNode` and `getNode` to return `KokkosClassic::DefaultNode`
   Mitigation: Remove all calls to `getDefaultNode` and `getNode`. `DefaultNode` objects are no longer used in Trilinos, so there is no need to get a `DefaultNode` object.
   Justification: `KokkosClassic::DefaultNode` is deprecated, along with functions that take a node object as an argument (see 3 and 4).

6. **Packages:** Tpetra, Xpetra, MueLu, Ifpack2
   Deprecation: `clone` methods for MultiVector, Vector, CrsGraph, CrsMatrix, Directory, Preconditioner
   Mitigation: Call copy constructors for the relevant classes; for deep copies, use `Teuchos::copy` argument.
   Justification: `KokkosClassic::DefaultNode` is deprecated, along with functions that take a node object as an argument. Thus, `clone` methods are no longer needed (see 3 and 4).

7. **Packages:** Tpetra
   Deprecation: `getDualView` method of `Tpetra::MultiVector`
   Mitigation: Use `getLocalViewDevice` or `getLocalViewHost`.
   Justification: Hides implementation details in Tpetra::MultiVector class.

8. **Packages:** Tpetra
   Deprecation: `Tpetra::MultiVector::normWeighted`
   Justification: Method was unused / untested in Trilinos; results are undefined for non-floating point types.

9. **Packages:** Tpetra
   Deprecation: `Tpetra::MultiVectorFiller`
   Mitigation: Use Tpetra::FEMultiVector for insertion / summation into off-processor multivector entries.
   Justification: Tpetra::FEMultiVector capabilities supercede Tpetra::MultiVectorFiller.

10. **Packages:** Tpetra, Ifpack2
    Deprecation: Methods `getGlobalNumDiags`, `getNodeNumDiags`, `isLowerTriangular`, `isUpperTriangular`
    Mitigation: Replicate mitigations implemented in Ifpack2; see `https://github.com/trilinos/Trilinos/issues/2630`.
    Justification: Removing these rarely used functions allows `CrsGraph` and `CrsMatrix` to avoid computation and global communication during construction

11. **Packages:** Tpetra
    Deprecation: `getLocalDiagOffsets` and `getLocalDiagCopy` overloads that return `Teuchos::ArrayView`
    Mitigation: Use overloads that return `Kokkos::View`
    Justification: `Kokkos::View` is compatible with device.

12. **Packages:** Tpetra
    Deprecation: Addition of two matrices with different row maps
    Mitigation: Export operand matrices to matrices that use identical row maps; the result matrix will share that row map.
    Justification: When matrices' row maps differ, the row map of the result matrix is not defined. Requiring users to provide matrices with the same row maps clearly defines the row map of the result matrix.

13. **Packages:** Tpetra
    Deprecation: `copyAndPermute`, `packAndPrepare`, `unpackAndCombine`, `doTransfer` using `Teuchos::ArrayView` arguments; `copyAndPermuteNew`, `packAndPrepareNew`, `unpackAndCombineNew`, `doTransferNew`, `useNewInterface`
    Mitigation: Update calls to `copyAndPermute`, `packAndPrepare`, `unpackAndCombine`, `doTransfer` to use new `Kokkos::View` arguments. Replace calls to `copyAndPermuteNew`, `packAndPrepareNew`, `unpackAndCombineNew`, `doTransferNew` with calls to `copyAndPermute`, `packAndPrepare`, `unpackAndCombine`, `doTransfer`. In user-defined objects inheriting from `Tpetra::DistObject`, implement interface with `Kokkos::View` arguments.
    Justification: To pack/unpack data on device, `Kokkos::View` arguments are needed; `Teuchos::ArrayView` requires data copy to host.

14. **Packages:** Tpetra
    Deprecation: Test-helper method `generateMatrix`
    Mitigation: Use matrix generation package like Galeri for tests.
    Justification: This code is unused and untested in Tpetra.

15. **Packages:** Tpetra
    Deprecation: `Tpetra::MatrixMarket` typedefs `comm_ptr`, `map_ptr`, `node_ptr`
    Mitigation: Replace `comm_ptr` and `map_ptr` with `Teuchos::RCP<const Teuchos::Comm<int> >` and `Teuchos::RCP<const Tpetra::Map<lno_t, gno_t>>`. `node_ptr` is no longer relevant.
    Justification: These public typedefs were unused in the MatrixMarket class.

16. **Packages:** Teuchos
    Deprecation: `PerformanceMonitorBase::clearTimers` method
    Mitigation: Use `PerformanceMonitorBase::clearCounters`.
    Justification: `PerformanceMonitorBase::clearTimers` is inaccurately named; data are not necessarily timers.

17. **Packages:** Teuchos
Deprecation: `count()` method in ArrayRCP, RCP, RCPNode
Mitigation: Use `strong_count()` methods
Justification: `count()` is redundant with `strong_count()`.

18. **Packages:** Teuchos
Deprecation: `LAPACK::POSVX`, `LAPACK::GESVX` methods that take `EQUED` by value
Mitigation: Pass pointer to EQUED.
Justification: LAPACK can change EQUED.

19. **Packages:** Teuchos
Deprecation: `LAPACK::TREXC` method that takes `ifst` and `ilst` by value
Mitigation: Pass pointers to ifst and ilst.
Justification: LAPACK can change ifst and ilst.

20. **Packages:** Teuchos
Deprecation: `LAPACK::GEBAL` method that takes `ilo` and `ihi` by value
Mitigation: Pass pointers to ilo and ihi.
Justification: LAPACK can change ilo and ihi.

21. **Packages:** Teuchos
Deprecation: `ArrayArg` class that creates an array of arguments
Mitigation: Pass individual arguments to functions
Justification: Remove unused, untested class providing trivial capabilities.

22. **Packages:** Teuchos
Deprecation: `MPITraits` class
Mitigation: Not applicable
Justification: Remove unused, untested class.

23. **Packages:** Teuchos
Deprecation: Operator overload of `<<` in `SerialDenseMatrix`, `SerialBandDenseMatrix`, `SerialSymDenseMatrix`, `SerialDenseVector`
Mitigation: Use method `print()`; e.g., `cout << A.print(cout) << endl;`.
Justification: Enable applications to provide their own implementations of overloaded ostream operator.

24. **Packages:** Belos
Deprecation: `Belos::toString()`
Mitigation: Use `Belos::convertStatusTypeToString()`
Justification: Original name was a bad choice.

25. **Packages:** Claps, Globipack, Optipack Trios
Deprecation: Packages Claps, Globipack, Optipack and Trios are deprecated and will be removed.
Justification: These packages were unused and no longer supported.

26. **Packages:** Epetra
Deprecation: Class `Epetra_MpiSmpComm`
Justification: This experimental class is unused and untested.

27. **Packages:** MueLu
Deprecation: Classes `CreateEpetraPreconditioner`, `CreateTpetraPreconditioner`, `CreateXpetraPreconditioner` that take nullspace and/or coordinates directly as parameters
Mitigation: Set "Nullspace" and "Coordinates" in the parameter list first
Justification: The removed overloads were trivial wrappers that saved at most two lines of code.

28. **Packages:** Panzer
Deprecation: Method `getGlobalIndexer`
Mitigation: Call `getRangeGlobalIndexer` instead
Justification: Panzer is now differentiating range and domain global indexers.

29. **Packages:** Phalanx
Deprecation: Method `addDependentField` with non-const data
Mitigation: Use analogous method with const data
Justification: Dependent fields should not be changed in evaluators. Const protection prevents changes.

30. **Packages:** RTop
Deprecation: Global variable `show_spmd_apply_op_dump`

31. **Packages:** Tacho
Deprecation: `TACHO_USE_DEPRECATED_TASKSCHEDULER`, `TACHO_USE_DEPRECATED_TASKSCHEDULER_MULTIPLE`
Mitigation: None needed; these settings are off by default.
Justification: Relied on Kokkos::DeprecatedTaskScheduler

32. **Packages:** STK
Deprecation: `findPermutation`, `register_cell_topology`, `register_super_cell_topology`, `get_cell_topology`, and other functions
Mitigation: Contact STK developers for guidance
Justification: STK is an experimental package and is undergoing active development. As such there will be deprecations from time to time. STK's policy is to decorate deprecated code with the `STK_DEPRECATED` macro, and then wait at least 6 weeks before deletion.

33. **Packages:** Ifpack2
Deprecation: Preconditioner `Krylov` in namespace `Ifpack2::DeprecatedAndMayDisappearAtAnyTime`
Mitigation:
Justification:

34. **Packages:** Teuchos
Deprecation: Teuchos::Comm helpers `reduceAll` and `scan` that take pointers to return argument
Mitigation: Use versions of functions that take references to pointer.

35. **Packages:** Thyra
Deprecation: `getTestResults` method that takes pointers to arguments
Mitigation: Use versions of functions that take references to pointer.
Justification: Pointer safety.

36. **Packages:** Intrepid
Deprecation: All Kokkos related code in Intrepid is deprecated and needs to be explicitly enabled with the cmake option `Intrepid_ENABLE_DEPRECATED_KOKKOS_CODE=ON`
Mitigation: Use Intrepid2 for performance portability
Justification: Intrepid2 is the performance-portable implementation of Intrepid. The Kokkos-related code still present in Intrepid is a leftover from a preliminary and no longer used performance-portable implementation.

37. **Packages:** Intrepid2
Deprecation: Types `ordinal_view_type`, `ebasis_view_type`, `ecoordinates_view_type`, `ordinal_type_array_1d_host`, `ordinal_type_array_2d_host`, `ordinal_type_array_3d_host`, `ordinal_type_array_stride_1d_host`, `ordinal_type_array_`
`ordinal_type_array_2d`, `ordinal_type_array_3d`, `ordinal_type_array_stride_1d`, `outputViewType`, `pointViewType`, `scalarViewType`
Mitigation: Use corresponding CamelCase-named types: `OrdinalViewType`, `EBasisViewType`, `ECoordinatesViewType`, `OrdinalTypeArray1DHost`, `OrdinalTypeArray2DHost`, `OrdinalTypeArray3DHost`, `OrdinalTypeArrayStride1DHost`, `OrdinalTypeArray1D`, `OrdinalTypeArray2D`, `OrdinalTypeArray3D`, `OrdinalTypeArrayStride1D`, `OutputViewType`, `PointViewType`, `ScalarViewType`
Justification: Types were renamed to adopt the CamelCase naming convention

Testing without Deprecated code

To test your code against Trilinos without deprecated code, add the following flags to your CMake configuration command:

- Kokkos:
`-D KOKKOS_ENABLE_DEPRECATED_CODE=OFF`
- Tpetra:
`-D Tpetra_ENABLE_DEPRECATED_CODE=OFF`
- Belos:
`-D Belos_HIDE_DEPRECATED_CODE=ON`
- Epetra:
`-D Epetra_HIDE_DEPRECATED_CODE=ON`

- Ifpack2:
  `-D Ifpack2_HIDE_DEPRECATED_CODE=ON`
  `-D Ifpack2_ENABLE_DEPRECATED_CODE=OFF`
  `-D KOKKOS_ENABLE_DEPRECATED_CODE=OFF`
  `-D Tpetra_ENABLE_DEPRECATED_CODE=OFF`
- Intrepid:
  `-D Intrepid_ENABLE_DEPRECATED_KOKKOS_CODE=OFF`
- MueLu:
  `-D MueLu_ENABLE_DEPRECATED_CODE=OFF`
  `-D Tpetra_ENABLE_DEPRECATED_CODE=OFF`
  `-D KOKKOS_ENABLE_DEPRECATED_CODE=OFF`
- Panzer:
  `-D Panzer_HIDE_DEPRECATED_CODE=ON`
  `-D KOKKOS_ENABLE_DEPRECATED_CODE=OFF`
- Phalanx:
  `-D Phalanx_HIDE_DEPRECATED_CODE=ON`
- RTop:
  `-D RTop_HIDE_DEPRECATED_CODE=ON`
- STK:
  `-D STK_HIDE_DEPRECATED_CODE=ON`
- Tacho, Sacado, Stokhos:
  `-D KOKKOS_ENABLE_DEPRECATED_CODE=OFF`
- Teuchos:
  `-D Teuchos_HIDE_DEPRECATED_CODE=ON`
  `-D Tpetra_ENABLE_DEPRECATED_CODE=OFF`
- Thyra:
  `-D Thyra_HIDE_DEPRECATED_CODE=ON`
  `-D Tpetra_ENABLE_DEPRECATED_CODE=OFF`
- Xpetra:
  `-D Tpetra_ENABLE_DEPRECATED_CODE=OFF`
  `-D Xpetra_ENABLE_DEPRECATED_CODE=OFF`
- Claps, Globipack, Optipack, Trios:
  `-D Claps_HIDE_DEPRECATED_CODE=ON`
  `-D GlobiPack_HIDE_DEPRECATED_CODE=ON`
  `-D OptiPack_HIDE_DEPRECATED_CODE=ON`
  `-D Trios_HIDE_DEPRECATED_CODE=ON`

Github branches

Deprecated code will be removed via pull requests to the Trilinos develop branch. As these pull requests are created, users can build their applications against the pull request branches.

For example, the branch tpetra-remove-deprecated contains the anticipated Tpetra code that will be merged into the develop branch.

- ```
  git clone git@github.com:trilinos/Trilinos
  cd Trilinos
  git checkout tpetra-remove-deprecated
  ```

Trilinos Product Owners

Karen Devine (kddevin@sandia.gov)
Roger Pawlowski (rppawlo@sandia.gov)
Mauro Perego (mperego@sandia.gov)
Siva Rajamanickam (srajama@sandia.gov)
Jim Willenbring (jmwille@sandia.gov)
  ```

Trilinos Management Contacts

Michael Wolf (mmwolf@sandia.gov)
Dena Vigil (dmvigi@sandia.gov)

SAND2019-12850 O