art news

Kyle J. Knoepfel

8 March 2018

art 2.11

- Features listed here: https://cdcvs.fnal.gov/redmine/projects/art/wiki/Series_211
- Most significant breaking change includes changes to `cet::map_vector<T>`
- Can insert elements in several ways

```
map_vector<string> mv;
mv[mv_key{3}] = "Alice";
mv[mv_key{7}] = "Erin";
mv[mv_key{5}] = "Charlie";
print(mv);
```

```
3: Alice
5: Charlie
7: Erin
```


Fermilab

art 2.11

- Features listed here: https://cdcvs.fnal.gov/redmine/projects/art/wiki/Series_211
- Most significant breaking change includes changes to `cet::map_vector<T>`
- Can insert elements in several ways

```
map_vector<string> mv;
mv[mv_key{3}] = "Alice";
mv[mv_key{7}] = "Erin";
mv[mv_key{5}] = "Charlie";

auto const pairs = {make_pair(mv_key{6}, "David"),
                    make_pair(mv_key{4}, "Bob")};
mv.insert(cbegin(pairs), cend(pairs));
print(mv);
```

Pre 2.11

```
3: Alice
5: Charlie
7: Erin
14: David
12: Bob
```


Fermilab

art 2.11

- Features listed here: https://cdcvs.fnal.gov/redmine/projects/art/wiki/Series_211
- Most significant breaking change includes changes to `cet::map_vector<T>`
- Can insert elements in several ways

```
map_vector<string> mv;
mv[mv_key{3}] = "Alice";
mv[mv_key{7}] = "Erin";
mv[mv_key{5}] = "Charlie";

auto const pairs = {make_pair(mv_key{6}, "David"),
                    make_pair(mv_key{4}, "Bob")};
mv.insert(cbegin(pairs), cend(pairs));
print(mv);
```

2.11

```
3: Alice
5: Charlie
6: David
4: Bob
7: Erin
```

Keys no longer adjusted

Fermilab

art 2.11

- Features listed here: https://cdcvs.fnal.gov/redmine/projects/art/wiki/Series_211
- Most significant breaking change includes changes to `cet::map_vector<T>`
- Can insert elements in several ways

```
map_vector<string> mv;
mv[mv_key{3}] = "Alice";
mv[mv_key{7}] = "Erin";
mv[mv_key{5}] = "Charlie";

auto const pairs = {make_pair(mv_key{6}, "David"),
                    make_pair(mv_key{4}, "Bob")};
mv.insert(cbegin(pairs), cend(pairs));
print(mv);
```

2.11

```
3: Alice
5: Charlie
6: David
4: Bob
7: Erin
```

Precondition: pairs to insert are sorted (bug/feature request)

Fermilab

art 2.11

- Features listed here: https://cdcvs.fnal.gov/redmine/projects/art/wiki/Series_211
- Most significant breaking change includes changes to `cet::map_vector<T>`
- Can insert elements in several ways

```
map_vector<string> mv;
mv[mv_key{3}] = "Alice";
mv[mv_key{7}] = "Erin";
mv[mv_key{5}] = "Charlie";

auto const pairs = {make_pair(mv_key{4}, "Bob"),
                    make_pair(mv_key{6}, "David")};
mv.insert(cbegin(pairs), cend(pairs));
print(mv);
```

2.11

```
3: Alice
4: Bob
5: Charlie
6: David
7: Erin
```

Precondition: pairs to insert are sorted (bug/feature request)

Fermilab

art 2.11

- Features listed here: https://cdcvs.fnal.gov/redmine/projects/art/wiki/Series_211
- Most significant breaking change includes changes to `cet::map_vector<T>`
- Can insert elements in several ways

```
map_vector<string> mv;
mv[mv_key{3}] = "Alice";
mv[mv_key{7}] = "Erin";
mv[mv_key{5}] = "Charlie";

auto const pairs = {make_pair(mv_key{6}, "David"),
                    make_pair(mv_key{4}, "Bob")};
for (auto const& elem : pairs) {
  mv.push_back(elem);
}
print(mv);
```

Pre 2.11

```
3: Alice
5: Charlie
7: Erin
14: David
19: Bob
```


Fermilab

art 2.11

- Features listed here: https://cdcvs.fnal.gov/redmine/projects/art/wiki/Series_211
- Most significant breaking change includes changes to `cet::map_vector<T>`
- Can insert elements in several ways

```
map_vector<string> mv;
mv[mv_key{3}] = "Alice";
mv[mv_key{7}] = "Erin";
mv[mv_key{5}] = "Charlie";

auto const pairs = {make_pair(mv_key{6}, "David"),
                    make_pair(mv_key{4}, "Bob")};
for (auto const& elem : pairs) {
  mv.insert(elem);
}
print(mv);
```

2.11

```
3: Alice
4: Bob
5: Charlie
6: David
7: Erin
```

Keys no longer adjusted

Fermilab

art 2.11

- Features listed here: https://cdcvs.fnal.gov/redmine/projects/art/wiki/Series_211
- Most significant breaking change includes changes to `cet::map_vector<T>`
- Keys are not adjusted by any insertion mechanism
- Key-adjustment occurs only during product-mixing, where adding offsets to the keys is necessary to avoid repeated (or elided) entries
  - No change relative to *art* 2.10 and older

Fermilab

art 3.0 status

- Home stretch
- Data-dependency errors can now be caught if you use the consumes interface
  - Some interface yet to be developed
  - Other loose ends
  - More testing
  - More documentation

Fermilab

Plans for C++17

- Modern C++ now means C++17
  - The C++17 standard was published in December 2017.
- The *art* team has provided C++17 support
  - **`c2`** qualifier (Clang 5.0.1 + GCC 6.4) – *art* 2.10 in January 2018
  - **`e17`** qualifier (GCC 7.3) – *art* 2.11 in April 2018
- Tentative plan:
  - after *art* 3.00 is released, we will support only C++17 builds beginning with *art* 3.01.
  - *art* code itself would not use any C++17 features until at least *art* 3.02
- The plan is subject to change based on release schedules and user input.

Fermilab