

Eigen::DenseStorage
< T, Dynamic, _Rows,
Dynamic, _Options >
::resize

Eigen::DenseStorage
< T, Dynamic, Dynamic,
_Cols, _Options >::resize

Eigen::DenseStorage
< T, Dynamic, Dynamic,
Dynamic, _Options >::
resize

Eigen::TensorStorage
< T, DSizes< IndexType,
NumIndices_ >, Options
_ >::resize

Eigen::TensorStorage
< T, DSizes< IndexType,
NumIndices_ >, Options
_ >::TensorStorage

Eigen::internal::conditional
_aligned_new_auto

```
graph LR; A["Eigen::DenseStorage< T, Dynamic, _Rows, Dynamic, _Options >::resize"] --> E["Eigen::internal::conditional_aligned_new_auto"]; B["Eigen::DenseStorage< T, Dynamic, Dynamic, _Cols, _Options >::resize"] --> E; C["Eigen::DenseStorage< T, Dynamic, Dynamic, Dynamic, _Options >::resize"] --> E; D["Eigen::TensorStorage< T, DSizes< IndexType, NumIndices_ >, Options_ >::resize"] --> E; F["Eigen::TensorStorage< T, DSizes< IndexType, NumIndices_ >, Options_ >::TensorStorage"] --> E;
```

The diagram illustrates the relationship between various Eigen storage classes and a common internal function. On the left, five boxes represent different storage classes: three variants of `Eigen::DenseStorage` and two variants of `Eigen::TensorStorage`. Each box contains a `resize` method call. Blue arrows point from each of these five boxes to a single, shaded box on the right labeled `Eigen::internal::conditional_aligned_new_auto`. This indicates that all these storage classes delegate their `resize` operation to this internal function.