

```
Eigen::internal::qr
_preconditioner_impl
< MatrixType, ColPivHouseholder
QRPreconditioner, PreconditionIfMore
ColsThanRows, true >::allocate
```

```
Eigen::internal::qr
_preconditioner_impl
< MatrixType, ColPivHouseholder
QRPreconditioner, PreconditionIfMore
RowsThanCols, true >::allocate
```

```
Eigen::internal::qr
_preconditioner_impl
< MatrixType, FullPivHouseholder
QRPreconditioner, PreconditionIfMore
ColsThanRows, true >::allocate
```

```
Eigen::internal::qr
_preconditioner_impl
< MatrixType, FullPivHouseholder
QRPreconditioner, PreconditionIfMore
RowsThanCols, true >::allocate
```

```
Eigen::internal::qr
_preconditioner_impl
< MatrixType, HouseholderQRPreconditioner,
PreconditionIfMoreColsThanRows,
true >::allocate
```

```
Eigen::internal::qr
_preconditioner_impl
< MatrixType, HouseholderQRPreconditioner,
PreconditionIfMoreRowsThanCols,
true >::allocate
```

```
Eigen::JacobiSVD::JacobiSVD
```

```
Eigen::JacobiSVD::rows
```

```
graph LR; A["Eigen::internal::qr_preconditioner_impl< MatrixType, ColPivHouseholderQRPreconditioner, PreconditionIfMore ColsThanRows, true >::allocate"] --> F["Eigen::JacobiSVD::rows"]; B["Eigen::internal::qr_preconditioner_impl< MatrixType, ColPivHouseholderQRPreconditioner, PreconditionIfMore RowsThanCols, true >::allocate"] --> F; C["Eigen::internal::qr_preconditioner_impl< MatrixType, FullPivHouseholderQRPreconditioner, PreconditionIfMore ColsThanRows, true >::allocate"] --> F; D["Eigen::internal::qr_preconditioner_impl< MatrixType, FullPivHouseholderQRPreconditioner, PreconditionIfMore RowsThanCols, true >::allocate"] --> F; E["Eigen::internal::qr_preconditioner_impl< MatrixType, HouseholderQRPreconditioner, PreconditionIfMoreColsThanRows, true >::allocate"] --> F; G["Eigen::internal::qr_preconditioner_impl< MatrixType, HouseholderQRPreconditioner, PreconditionIfMoreRowsThanCols, true >::allocate"] --> F; H["Eigen::JacobiSVD::JacobiSVD"] --> F; style F fill:#808080,color:#fff
```