

```
Eigen::internal::evaluator  
< Solve< Decomposition,  
  RhsType > >::evaluator
```

```
Eigen::internal::Assignment  
< DstXprType, Solve< CwiseUnary  
Op< internal::scalar_conjugate  
_op< typename DecType::Scalar  
>, const Transpose< const DecType  
> >, RhsType >, internal::assign  
_op< Scalar, Scalar >, Dense2Dense >::run
```

```
Eigen::internal::Assignment  
< DstXprType, Solve< DecType,  
  RhsType >, internal::assign  
_op< Scalar, Scalar >, Dense2Dense >::run
```

```
Eigen::internal::Assignment  
< DstXprType, Solve< DecType,  
  RhsType >, internal::assign  
_op< Scalar, Scalar >, Sparse2Sparse >::run
```

```
Eigen::internal::Assignment  
< DstXprType, Solve< Transpose  
< const DecType >, RhsType >  
, internal::assign_op< Scalar,  
  Scalar >, Dense2Dense >::run
```

Eigen::Solve::rhs

```
graph LR; A[Eigen::internal::evaluator< Solve< Decomposition, RhsType > >::evaluator] --> E[Eigen::Solve::rhs]; B[Eigen::internal::Assignment< DstXprType, Solve< CwiseUnaryOp< internal::scalar_conjugate_op< typename DecType::Scalar >, const Transpose< const DecType > >, RhsType >, internal::assign_op< Scalar, Scalar >, Dense2Dense >::run] --> E; C[Eigen::internal::Assignment< DstXprType, Solve< DecType, RhsType >, internal::assign_op< Scalar, Scalar >, Dense2Dense >::run] --> E; D[Eigen::internal::Assignment< DstXprType, Solve< DecType, RhsType >, internal::assign_op< Scalar, Scalar >, Sparse2Sparse >::run] --> E; F[Eigen::internal::Assignment< DstXprType, Solve< Transpose< const DecType >, RhsType >, internal::assign_op< Scalar, Scalar >, Dense2Dense >::run] --> E;
```