

```
Eigen::internal::make  
_coherent_impl< A, Matrix  
< B_Scalar, B_Rows, B_Cols,  
B_Options, B_MaxRows, B_MaxCols  
> >::run
```

```
Eigen::internal::make  
_coherent_impl< Matrix  
< A_Scalar, A_Rows, A_Cols,  
A_Options, A_MaxRows, A_MaxCols  
>, B >::run
```

```
Eigen::internal::make  
_coherent_expression
```

```
Eigen::internal::Assignment  
< DstXprType, CwiseBinaryOp  
< internal::scalar_product  
_op< ScalarBis, Scalar >, const  
CwiseNullaryOp< internal::scalar  
_constant_op< ScalarBis >, Plain  
>, const Product< Lhs, Rhs, DefaultProduct  
> >, AssignFunc, Dense2Dense >::run
```

```
Eigen::internal::check  
_transpose_aliasing_run  
_time_selector< Scalar,  
DestIsTransposed, CwiseBinaryOp  
< BinOp, DerivedA, DerivedB > >::run
```

```
Eigen::CwiseBinaryOp::rhs
```

```
graph LR; A["Eigen::internal::make  
_coherent_impl< A, Matrix  
< B_Scalar, B_Rows, B_Cols,  
B_Options, B_MaxRows, B_MaxCols  
> >::run"] --> B["Eigen::internal::make  
_coherent_expression"]; C["Eigen::internal::make  
_coherent_impl< Matrix  
< A_Scalar, A_Rows, A_Cols,  
A_Options, A_MaxRows, A_MaxCols  
>, B >::run"] --> B; B --> D["Eigen::internal::Assignment  
< DstXprType, CwiseBinaryOp  
< internal::scalar_product  
_op< ScalarBis, Scalar >, const  
CwiseNullaryOp< internal::scalar  
_constant_op< ScalarBis >, Plain  
>, const Product< Lhs, Rhs, DefaultProduct  
> >, AssignFunc, Dense2Dense >::run"]; D --> E["Eigen::CwiseBinaryOp::rhs"]; F["Eigen::internal::check  
_transpose_aliasing_run  
_time_selector< Scalar,  
DestIsTransposed, CwiseBinaryOp  
< BinOp, DerivedA, DerivedB > >::run"] --> E; style E fill:#808080
```