

## Avoid virtual calls between nodes for non-jitted transformations

```
TInterface<TFilterBase> d2 = d.Filter(filterFun)
```

```
TInterface<TFilterBase> d2 = d.Filter("fiterString")
```

## Avoid virtual calls between nodes for non-jitted transformations

```
TInterface<TFilter>      d2 = d.Filter(filterFun)
```

```
TInterface<TFilterBase> d2 = d.Filter("fiterString")
```

“Only” requires **changing the return type** of Filter, Define, Range in the non-jitted case.

## Avoid virtual calls between nodes for non-jitted transformations

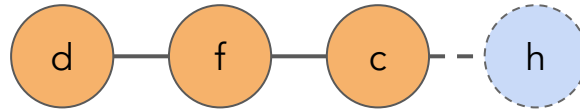
```
auto d2 = d.Filter(filterFun)
```

```
auto d2 = d.Filter("fiterString")
```

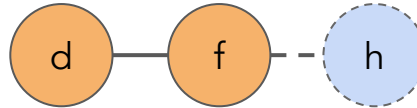
...should not be a problem as we expect users to never write down those types explicitly

# Remove "defines" from the graph w/o changing visible behaviour

```
TInterface<TCustomColumn> d2 = d.Filter().Define("x")
```



```
TInterface<Filter> d2 = d.Filter().Define("x")
```

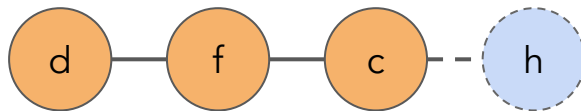


# Remove "defines" from the graph w/o changing visible behaviour

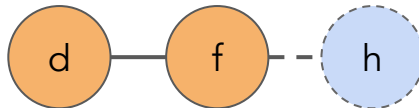
`TCustomColumn::validBranches == {..., "x"}`

`SomeNode::validBranches == {...}`

`TInterface<TCustomColumn> d2 = d.Filter().Define("x")`



`TInterface<Filter> d2 = d.Filter().Define("x")`

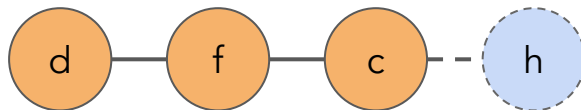


# Remove "defines" from the graph w/o changing visible behaviour

`TCustomColumn::validBranches == {..., "x"}`

`SomeNode::validBranches == {...}`

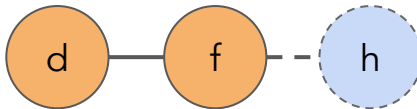
`TInterface<TCustomColumn> d2 = d.Filter().Define("x")`



`TInterface::validBranches == {..., "x"}`

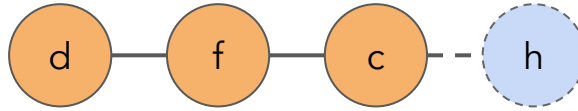
`TInterface::validBranches == {...}`

`TInterface<Filter> d2 = d.Filter().Define("x")`

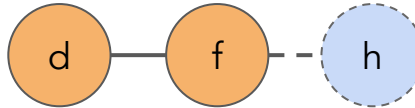


# Remove "defines" from the graph w/o changing visible behaviour

```
auto d2 = d.Filter().Define("x")
```



```
auto d2 = d.Filter().Define("x")
```



Again this should not matter to users as they should not care about node types, only about their behaviour.