

COMONAD TRANSFORMERS

COBOTS IN DISGUISE

COMONAD TRANSFORMERS

STACK

JUST LIKE MONAD TRANSFORMERS

AN ENVIRONMENT WITH A STORE INSIDE

EnvT Settings (Store Int) Double

JUST LIKE MONAD TRANSFORMERS
WE CAN NOW ACCESS THE INTERFACE
FOR ANY COMONAD IN THE STACK

A FEW EXAMPLES

```
ask    :: ComonadEnv   e w =>           w a -> e
trace :: ComonadTraced m w => m -> w a -> a
peek  :: ComonadStore  s w => s -> w a -> a
```

```
class ComonadTrans t where  
    lower :: Comonad w => t w a -> w a
```

MONAD TRANSFORMERS LIFT
COMONAD TRANSFORMERS LOWER

YOU CAN PEEL OFF
LAYERS
FROM THE TOP DOWN

ENVT

```
data EnvT e w a = EnvT e (w a)
```

```
lower (EnvT e w) = w
```

STORET

```
data StoreT s w a = StoreT (w (s -> a)) s
```

```
lower (StoreT wf s) = fmap ($ s) wf
```

TRACEDT

```
newtype TracedT m w a = TracedT (w (m -> a))  
lower (TracedT wf) = fmap ($ mempty) wf
```

EXTRACT ON A TRANSFORMER GOES ALL THE WAY DOWN

EXAMPLE

```
λ> let t = EnvT "hi" (traced (\(Sum x) -> x + 10))
```

```
t :: EnvT "hi" (Traced (Sum Int)) Int
```

```
λ> extract t
```

```
10
```

```
λ> trace (Sum 1) t
```

```
11
```

```
λ> ask t
```

```
"hi"
```

LET'S WRITE
A REPORT!

WE'LL USE AN ENVIRONMENT
TO SPECIFY WHETHER WE WANT IT TO BE
DETAILED OR A SUMMARY

```
data ReportStyle = Detailed | Summary
```

```
reportConfig :: EnvT ReportStyle w a
```

```
reportConfig = EnvT Summary _
```

WE'LL USE A TRACED COMONAD
TO TRACK WHICH MONTH
THE REPORT IS FOR

SETUP

```
-- Track sales projections over coming months
projections :: Sum Int -> Float
projections (Sum month) = 1.2 ^ (max 0 month) * 100

reportConfig :: EnvT ReportStyle (Traced (Sum Int)) a
reportConfig = EnvT Detailed (traced projections)
```

SOME QUERIES

```
previousMonth :: ComonadTraced (Sum Int) w => w a -> a  
previousMonth = trace (Sum (-1))
```

```
nextMonth :: ComonadTraced (Sum Int) w => w a -> a  
nextMonth = trace (Sum 1)
```

```
previousMonth = trace (Sum (-1))
```

```
nextMonth = trace (Sum 1)
```

```
λ> reportConfig & extract
```

```
100.0
```

```
λ> reportConfig =>> nextMonth & extract
```

```
120.00001
```

```
λ> reportConfig =>> nextMonth =>> nextMonth =>> previousMonth & extract
```

```
120.00001
```

MAKE A SIMPLE REPORT

```
detailedReport :: (ComonadTraced (Sum Int) w) => w Float -> String
detailedReport = do
    salesAmt <- extract
    prev <- previousMonth
    next <- nextMonth
    return $ unlines [ "This months sales in totality are: " <>> show salesAmt
                    , "Previous month's sales: " <>> show prev
                    , "Next month's projections: " <>> show next
                    ]
```

```
buildHeader :: (ComonadEnv ReportStyle w) => w a-> String
buildHeader = do
    style <- ask
    return $ case style of
        Detailed -> "Please find enclosed your DETAILED report: \n"
        Summary -> "Please find enclosed your SUMMARY report: \n"
```

```
λ> buildHeader reportConfig  
"Please find enclosed your DETAILED report\n"
```

```
buildReport :: (ComonadTraced (Sum Int) w, ComonadEnv ReportStyle w)
              => w Float -> String
buildReport = do
  header <- buildHeader
  salesAmt <- extract
  style <- ask
  case style of
    Summary ->
      return $ header <> "We achieved " <> show salesAmt <> " in sales!"
    Detailed -> do
      rpt <- detailedReport
      return $ header <> rpt
```

```
λ> putStrLn $ buildReport reportConfig  
Please find enclosed your DETAILED report:  
This months sales in totality are: 100.0  
Previous month's sales: 100.0  
Next month's projections: 120.0
```

```
λ> putStrLn $ reportConfig =>> nextMonth =>> buildReport & extract  
Please find enclosed your DETAILED report:  
This months sales in totality are: 120.0  
Previous month's sales: 100.0  
Next month's projections: 144.0
```

```
λ> putStrLn $ reportConfig =>> nextMonth =>> buildReport . local (const Summary) & extract  
Please find enclosed your SUMMARY report:  
We achieved 120.0 in sales!
```

QUESTIONS?

LET'S ADD
ONE MORE LAYER

WHAT IF WE WANT A REPORT FOR EACH REGION

```
data Region = America | UK | Germany  
deriving (Show, Eq, Ord)
```

```
projections :: Region -> Sum Int -> Float  
projections UK (Sum month) = 1.2 ^ (max 0 month) * 100  
projections America (Sum month) = 1.3 ^ (max 0 month) * 200  
projections Germany (Sum month) = 1.5 ^ (max 0 month) * 300
```

JUST PLAY TYPE TETRIS FOR THIS STUFF...

```
data EnvT e w a = EnvT e (w a)
newtype TracedT m w a = TracedT (w (m -> a))
data StoreT s w a = StoreT (w (s -> a)) s
```

```
reportConfig
  :: EnvT ReportStyle (TracedT (Sum Int) (Store Region)) Float
reportConfig = (EnvT Detailed (TracedT (store projections UK)))
```

NOW THAT WE HAVE A STORE
WE CAN EXPERIMENT IN OTHER REGIONS

```
otherRegions :: (ComonadStore Region w) => w a -> [a]
otherRegions w = experiment others w
where
    others currentRegion = filter (/= currentRegion) allRegions
```

```
allRegions :: [Region]
allRegions = [UK, America, Germany]
```

```
projections UK      (Sum month) = 1.2 ^ (max 0 month) * 100
projections America (Sum month) = 1.3 ^ (max 0 month) * 200
projections Germany (Sum month) = 1.5 ^ (max 0 month) * 300
```

```
λ> extract reportConfig
100.0
```

```
-- others UK == [America, Germany]
λ> otherRegions reportConfig
[200.0,300.0]
```

USING BOTH STORE
AND TRACED
WE CAN BUILD REGIONAL REPORTS FOR
OTHER MONTHS!

```
λ> reportConfig & otherRegions  
[200.0,300.0]
```

```
λ> reportConfig =>> otherRegions =>> trace 3 & extract  
[439.39996,1012.5]
```

```
λ> reportConfig =>> trace 3 =>> otherRegions & extract  
[439.39996,1012.5]
```

```
λ> reportConfig =>> trace 1 =>> otherRegions =>> trace 2 & extract  
[439.39996,1012.5]
```

EXTENDING TRACE SHIFTS US TO DIFFERENT VIEWS

EVERY VIEW IS KEPT UP TO DATE

LAZY

```
-- otherRegions :: (ComonadStore Region w) => w a -> [a]

comparisonReport :: (ComonadTraced (Sum Int) w, ComonadStore Region w)
                  => w Float -> String
comparisonReport w =
  let otherReports = w =>> detailedReport =>> otherRegions & extract
  in "Comparison Report\n" <> unlines otherReports
```

```
λ> putStrLn $ comparisonReport reportConfig
```

Comparison Report

America:

This months sales in totality are: 200.0

Previous month's sales: 200.0

Next month's projections: 260.0

Germany:

This months sales in totality are: 300.0

Previous month's sales: 300.0

Next month's projections: 450.0

```
buildReport :: (ComonadTraced (Sum Int) w, ComonadEnv ReportStyle w, ComonadStore Region w)
              => w Float -> String
buildReport = do
  header <- buildHeader
  salesAmt <- extract
  style <- ask
  case style of
    Summary -> return $ header <> "We achieved " <> show salesAmt <> " in sales!"
    Detailed -> do
      rpt <- detailedReport
      compReport <- comparisonReport
      return $ header <> rpt <> "\n" <> compReport
```

```
λ> putStrLn $ buildReport reportConfig
Please find enclosed your DETAILED report
UK:
This months sales in totality are: 100.0
Previous month's sales: 100.0
Next month's projections: 120.00001
```

Comparison Report

America:

```
This months sales in totality are: 200.0
Previous month's sales: 200.0
Next month's projections: 260.0
```

Germany:

```
This months sales in totality are: 300.0
Previous month's sales: 300.0
Next month's projections: 450.0
```

```
λ> putStrLn $ reportConfig =>> trace 3 =>> buildReport & extract  
Please find enclosed your DETAILED report  
UK:  
This months sales in totality are: 172.80002  
Previous month's sales: 144.0  
Next month's projections: 207.36
```

Comparison Report
America:

```
This months sales in totality are: 439.39996  
Previous month's sales: 337.99997  
Next month's projections: 571.21985
```

Germany:

```
This months sales in totality are: 1012.5  
Previous month's sales: 675.0  
Next month's projections: 1518.75
```

```
λ> let monthlyReports = reportConfig =>> buildReport
λ> putStrLn $ peek Germany monthlyReports
Please find enclosed your DETAILED report
Germany:
This months sales in totality are: 300.0
Previous month's sales: 300.0
Next month's projections: 450.0
```

Comparison Report

UK:

```
This months sales in totality are: 100.0
Previous month's sales: 100.0
Next month's projections: 120.00001
```

America:

```
This months sales in totality are: 200.0
Previous month's sales: 200.0
Next month's projections: 260.0
```

```
λ> let monthlyReports = reportConfig =>> buildReport  
λ> let germanyReport = peek Germany monthlyReports  
λ> let germanyMarchReport = monthlyReports =>> peek Germany =>> trace 3 & extract  
λ> let allFebruaryReports = monthlyReports =>> trace 2 =>> experiment (const allRegions)
```

chrисpenner.ca

github.com/ChrisPenner



@ChrisLPenner