How to Beautify SQL and PL/SQL Code

Philipp

@phsalvisberg

https://www.salvis.com/blog

17.11.2020

Philipp

- Database centric development
- Model Driven Software Development
- Author of free SQL Developer Extensions PL/SQL Unwrapper, PL/SQL Cop, utPLSQL, plscope-utils, oddgen and Bitemp Remodeler

@phsalvisberg

https://www.salvis.com/blog

trivadis

FOUNDED IN
1994

15 TRIVADIS WORKSPACES

SWITZERLAND, GERMANY, AUSTRIA, ROMANIA

118 MILLION CHF
TURNOVER

300 SLA's
(SERVICE LEVEL AGREEMENTS)

4000
TRAINING PARTICIPANTS PER YEAR

800
CUSTOMERS

700
EMPLOYEES

5 MILLION CHF
BUDGET FOR SCIENCE AND DEVELOPMENT PER YEAR

EXPERIENCE FROM
1900 PROJECTS
PER YEAR

trivadis

Syntax Highlighting

Code as Plain Text

trivadis

```
BEGIN
   FOR rec IN (
      SELECT r.country_region AS region, p.prod_category, SUM(s.amount_sold) AS amount_sold
        FROM sales     s
        JOIN products  p    ON p.prod_id    = s.prod_id
        JOIN customers cust ON cust.cust_id = s.cust_id
        JOIN times     t    ON t.time_id    = s.time_id
        JOIN countries r    ON r.country_id = cust.country_id
       WHERE calendar_year = 2000
       GROUP BY r.country_region, p.prod_category
       ORDER BY r.country_region, p.prod_category
   ) LOOP
      IF rec.region = 'Asia' THEN
         IF rec.prod_category = 'Hardware' THEN
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         END IF;
      END IF;
   END LOOP;
END;
/
```

Step 1 - Highlight Keywords

trivadis

```
BEGIN
   FOR rec IN (
      SELECT r.country_region AS region, p.prod_category, SUM(s.amount_sold) AS amount_sold
        FROM sales     s
        JOIN products  p    ON p.prod_id    = s.prod_id
        JOIN customers cust ON cust.cust_id = s.cust_id
        JOIN times     t    ON t.time_id    = s.time_id
        JOIN countries r    ON r.country_id = cust.country_id
       WHERE calendar_year = 2000
       GROUP BY r.country_region, p.prod_category
       ORDER BY r.country_region, p.prod_category
   ) LOOP
      IF rec.region = 'Asia' THEN
         IF rec.prod_category = 'Hardware' THEN
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         END IF;
      END IF;
   END LOOP;
END;
/
```

UPPERCASE?

trivadis

Source: https://twitter.com/TomCostantino/status/1196257484370935808/photo/1

Step 2 – lowercase Keywords

trivadis

```
begin
   for rec in (
      select r.country_region as region, p.prod_category, sum(s.amount_sold) as amount_sold
        from sales     s
        join products  p    on p.prod_id    = s.prod_id
        join customers cust on cust.cust_id = s.cust_id
        join times     t    on t.time_id    = s.time_id
        join countries r    on r.country_id = cust.country_id
       where calendar_year = 2000
       group by r.country_region, p.prod_category
       order by r.country_region, p.prod_category
   ) loop
      if rec.region = 'Asia' then
         if rec.prod_category = 'Hardware' then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

Step 3 – Highlight Numbers

trivadis

```
begin
   for rec in (
      select r.country_region as region, p.prod_category, sum(s.amount_sold) as amount_sold
        from sales     s
        join products  p    on p.prod_id    = s.prod_id
        join customers cust on cust.cust_id = s.cust_id
        join times     t    on t.time_id    = s.time_id
        join countries r    on r.country_id = cust.country_id
       where calendar_year = 2000
       group by r.country_region, p.prod_category
       order by r.country_region, p.prod_category
   ) loop
      if rec.region = 'Asia' then
         if rec.prod_category = 'Hardware' then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

Step 3 – Highlight Strings

trivadis

```
begin
   for rec in (
      select r.country_region as region, p.prod_category, sum(s.amount_sold) as amount_sold
        from sales     s
        join products  p    on p.prod_id    = s.prod_id
        join customers cust on cust.cust_id = s.cust_id
        join times     t    on t.time_id    = s.time_id
        join countries r    on r.country_id = cust.country_id
       where calendar_year = 2000
       group by r.country_region, p.prod_category
       order by r.country_region, p.prod_category
   ) loop
      if rec.region = 'Asia' then
         if rec.prod_category = 'Hardware' then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

Step 4 – Highlight Comments

trivadis

```
begin
   for rec in (
      select r.country_region as region, p.prod_category, sum(s.amount_sold) as amount_sold
        from sales     s
        join products  p    on p.prod_id    = s.prod_id
        join customers cust on cust.cust_id = s.cust_id
        join times     t    on t.time_id    = s.time_id
        join countries r    on r.country_id = cust.country_id
       where calendar_year = 2000
       group by r.country_region, p.prod_category
       order by r.country_region, p.prod_category
   ) loop
      if rec.region = 'Asia' then
         if rec.prod_category = 'Hardware' then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

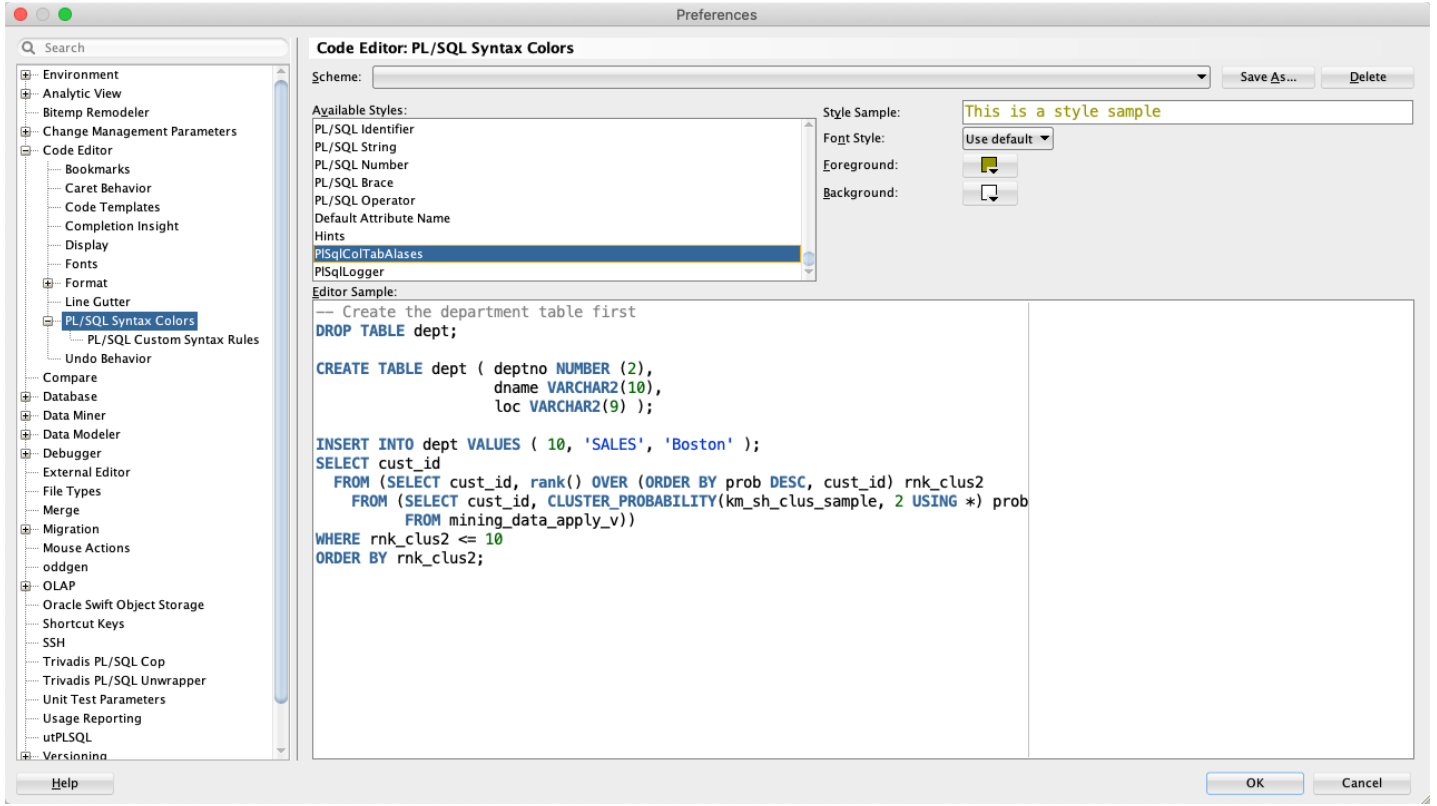
Step n – Highlight Aliases

trivadis

```
begin
   for rec in (
      select r.country_region as region, p.prod_category, sum(s.amount_sold) as amount_sold
        from sales     s
        join products  p    on p.prod_id    = s.prod_id
        join customers cust on cust.cust_id = s.cust_id
        join times     t    on t.time_id    = s.time_id
        join countries r    on r.country_id = cust.country_id
       where calendar_year = 2000
       group by r.country_region, p.prod_category
       order by r.country_region, p.prod_category
   ) loop
      if rec.region = 'Asia' then
         if rec.prod_category = 'Hardware' then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

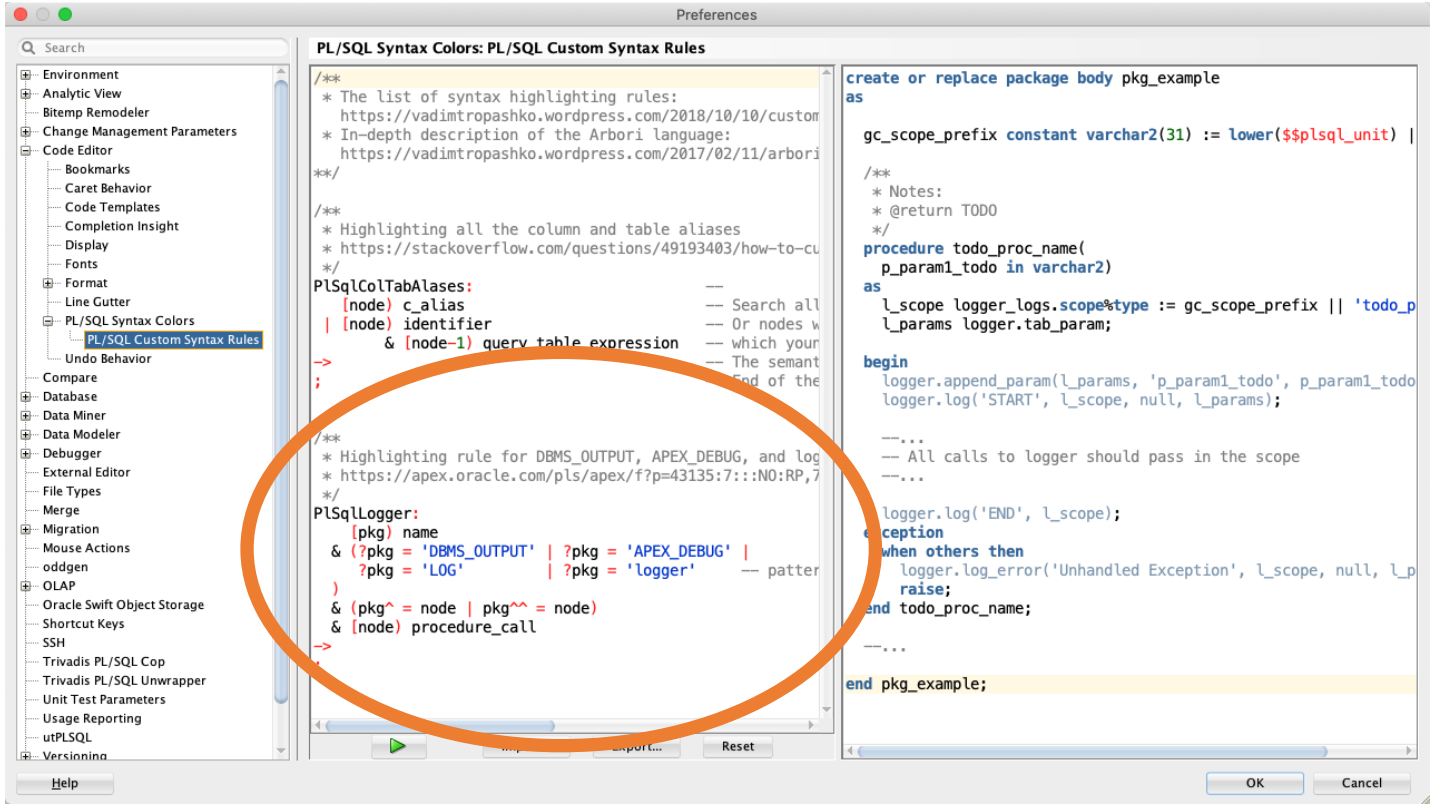
PL/SQL Syntax Colors

trivadis



PL/SQL Custom Syntax Rules

trivadis

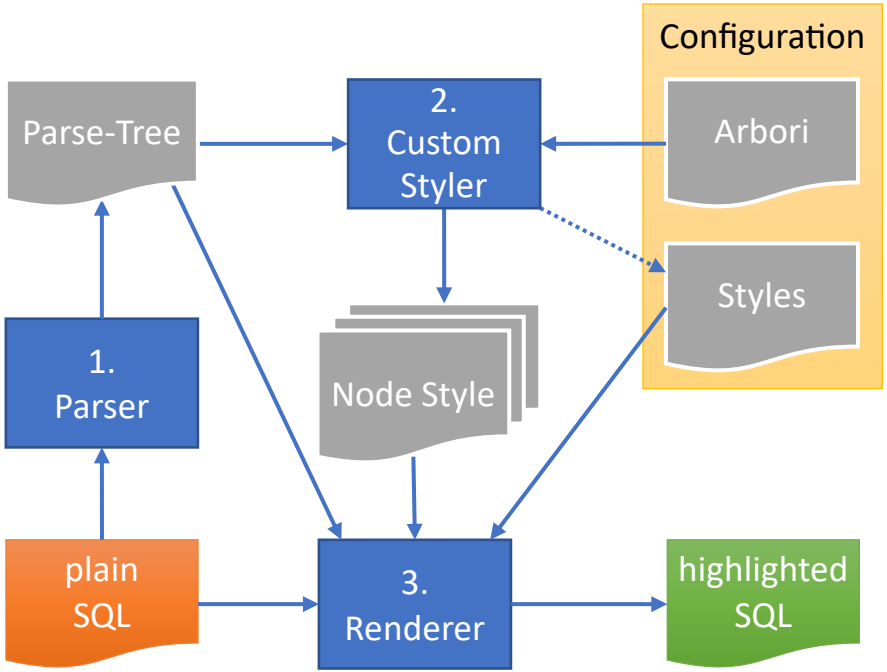


trivadis

Arbori – Part 1 (Styler)

How the Custom Styler (Highlighter) Works

trivadis



Java Callback Functions (->)

trivadis

```
public void PlSqlColTabAlases (Parsed target, Map<String, ParseNode> tuple) {
   addStyle(target, tuple.get("node"), "PlSqlColTabAlases");
}
```

```
public void PlSqlLogger (Parsed target, Map<String, ParseNode> tuple) {
   addStyle(target, tuple.get("node"), "PlSqlLogger");
}
```

JavaScript Callback Function – Example

trivadis

```
Operator:
  [node) ':' | [node) '=' | [node) '>' | [node) '<' | [node) '|' | [node) '!'
  -> {
     struct.addStyle(target, tuple.get("node"), "Operator");
  }
;
```

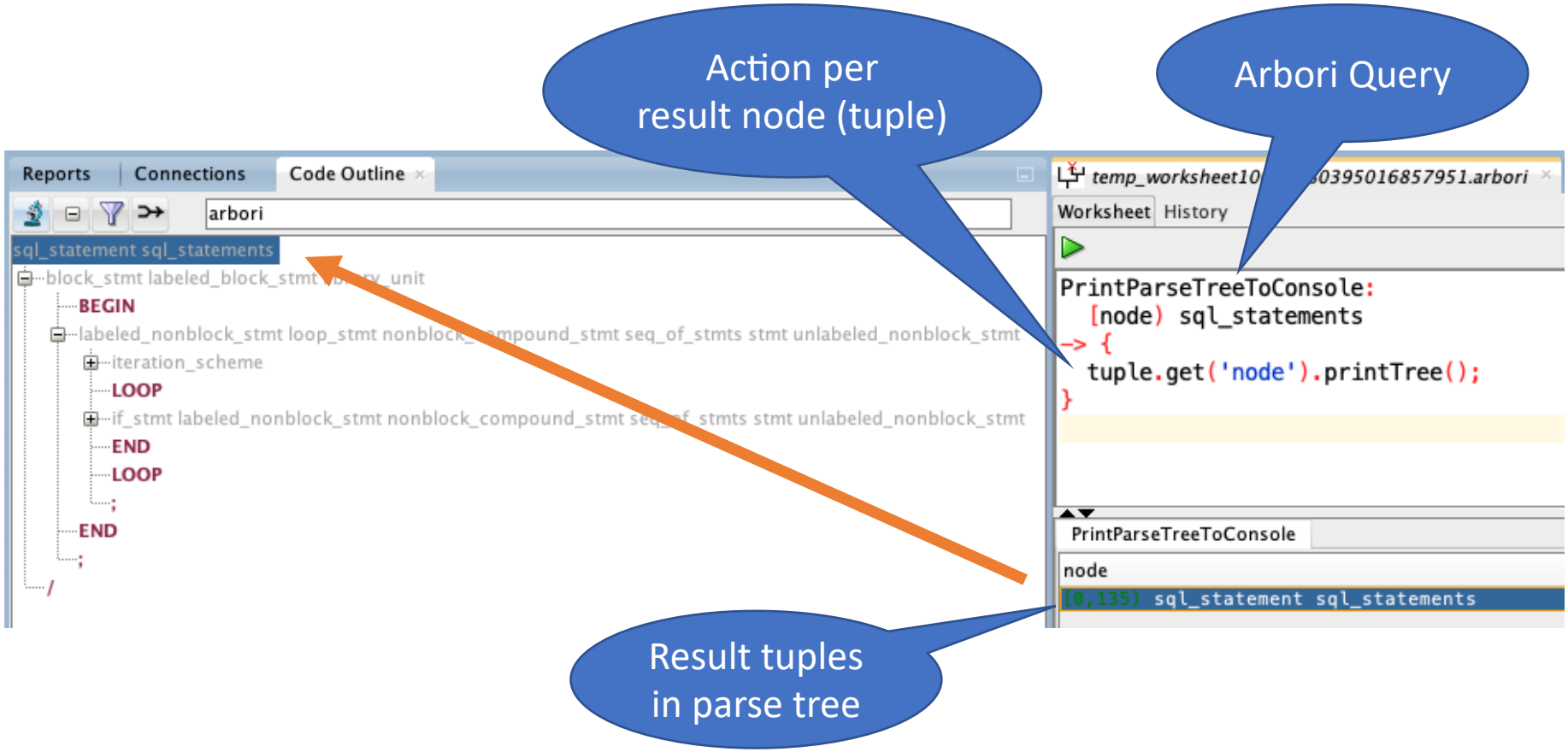
- Add Arbori query to "Custom Syntax Rules" (executed in the program's order)
- Restart SQL Developer
- "Operator" appears as new style under "PL/SQL Syntax Colors" (based on query name)
- Now you can change the font style (normal, bold, italic) and the foreground and background color

Code Outline in SQL Developer

trivadis

Arbori Editor

trivadis



trivadis

Demo 1

trivadis

Code Formatting (Beautifying)

Tokenized

trivadis

```
begin
for
rec
in
(
select
r
.
country_region
as
region
,
p
.
prod_category
,
sum
(
s
.
amount_sold
)
as
amount_sold
from
sales
s
join
products
p
on
p
.
prod_id
=
s
.
prod_id
join
customers
cust
on
cust
.
cust_id
=
s
.
cust_id
join
times
t
on
t
.
time_id
=
s
.
time_id
join
countries
r
on
r
.
country_id
=
cust
.
country_id
where
calendar_year
=
2000
group
by
r
.
country_region
,
p
.
prod_category
order
by
r
.
country_region
,
p
.
prod_category
)
loop
if
rec.region
=
'Asia'
then
if
rec
.
prod_category
=
'Hardware'
then
/* print only
one line for
demo purposes
*/
sys
.
dbms_output
.
put_line
(
'Amount: '
||
rec
.
amount_sold
)
;
end
if
;
end
if
;
end
loop;
end;
/
```

Minified – No Wrap Edition

trivadis

```
begin for rec in(select r.country_region as region,p.prod_category,sum(s.amount_sold)as amount_sold from sales s join products p on p.prod_id=s.prod_id join customers cust on cust.cust_id=s.cust_id join times t on t.time_id=s.time_id join countries r on r.country_id=cust.country_id where calendar_year=2000group by r.country_region,p.prod_category order by r.country_region,p.prod_category)loop if rec.region='Asia'then if rec.prod_category='Hardware'then/* print only one line for demo purposes */sys.dbms_output.put_line('Amount: '||rec.amount_sold );end if;end if;end loop;end;
/
```

Minified – Word Wrap Edition

trivadis

```
begin for rec in(select r.country_region as
region,p.prod_category,sum(s.amount_sold)as amount_sold from sales s join
products p on p.prod_id=s.prod_id join customers cust on cust.cust_id=s.cust_id
join times t on t.time_id=s.time_id join countries r on
r.country_id=cust.country_id where calendar_year=2000group by
r.country_region,p.prod_category order by r.country_region,p.prod_category)loop
if rec.region='Asia'then if rec.prod_category='Hardware'then/* print only one
line for demo purposes */sys.dbms_output.put_line('Amount: '||rec.amount_sold
);end if;end if;end loop;end;
/
```

Justified

trivadis

```
begin for rec in (select r.country_region as region, p.prod_category,
sum (s.amount_sold) as amount_sold from sales s join products p on p.
prod_id = s .prod_id join customers cust on cust .cust_id = s.cust_id
join times t on t.time_id= s.time_id join countries r on r.country_id
= cust .country_id where calendar_year=2000 group by r.country_region
, p . prod_category order by r . country_region , p . prod_category )
loop if rec.region = 'Asia'  then if rec . prod_category = 'Hardware'
then /* print only one line for demo purposes */  sys . dbms_output .
put_line('Amount: '||rec.amount_sold );end if; end if; end loop; end;
/
```

ASCII Art

trivadis

```
     begin                                  for rec
   in(select r                            .country_region
  as region ,p .                         prod_category,sum(
 s.amount_sold ) as                     amount_sold from sales
s join products p on                    p . prod_id = s .prod_id
join customers cust on                cust.cust_id=s.cust_id join
times t on t . time_id =s.      time_id join countries r on
  r.country_id = cust.country_id  where  calendar_year =
     2000 group by r.country_region , p.prod_category
       order by r .country_region, p.prod_category
         ) loop if rec . region = 'Asia' then if
           rec.prod_category = 'Hardware' then
             /* print only one line for demo
                purposes */sys.dbms_output
                  . put_line ( 'Amount: '
                    ||rec.amount_sold);
                      end if;end if;
                        end loop;
                          end;
                           /
```

Careless / Loveless

trivadis

```
begin
   for rec in (
   select r.country_region as region, p.prod_category, sum(s.amount_sold) as amount_sold
   from sales s join products p on p.prod_id = s.prod_id
   join customers cust on cust.cust_id = s.cust_id
   join times t on t.time_id = s.time_id
   join countries r
     on r.country_id = cust.country_id
   where calendar_year = 2000
   group by r.country_region, p.prod_category
   order by r.country_region, p.prod_category) loop
    if rec.region = 'Asia' then
if rec.prod_category = 'Hardware' then
   /* print only one line for demo purposes */
   sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
end if;
    end if;
  end loop;
end;
/
```

But the Machine Can Run This

trivadis

```
      begin                                  for rec
   in(select r                             .country_region
  as region ,p .                          prod_category,sum(
 s.amount_sold ) as                      amount_sold from sales
s join products p on                     p . prod_id = s .prod_id
join customers cust on                 cust.cust_id=s.cust_id join
times t on t . time_id =s.     time_id join countries r on
  r.country_id = cust.country_id  where  calendar_year =
     2000 group by r.country_region , p.prod_category
       order by r .country_region, p.prod_category
         ) loop if rec . region = 'Asia' then if
           rec.prod_category = 'Hardware' then
             /* print only one line for demo
                purposes */sys.dbms_output
                  . put_line ( 'Amount: '
                    ||rec.amount_sold);
                      end if;end if;
                        end loop;
                          end;
                           /
```

So What?

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."

-- Martin Fowler

Source: https://en.wikiquote.org/wiki/Martin_Fowler

"It is a matter of fact, that code is more often read than written – therefore we should take efforts to ease the work of the reader, which is not necessarily the author."

-- Roger Troller

Source: https://trivadis.github.io/plsql-and-sql-coding-guidelines/v3.6/

Readable – Style 1 of ∞

trivadis

```
begin
   for rec in (
      select r.country_region as region, p.prod_category, sum(s.amount_sold) as amount_sold
        from sales     s
        join products  p    on p.prod_id    = s.prod_id
        join customers cust on cust.cust_id = s.cust_id
        join times     t    on t.time_id    = s.time_id
        join countries r    on r.country_id = cust.country_id
       where calendar_year = 2000
       group by r.country_region, p.prod_category
       order by r.country_region, p.prod_category
   ) loop
      if rec.region = 'Asia' then
         if rec.prod_category = 'Hardware' then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

Readable – Style 2 of ∞

trivadis

```
begin
   for rec in (select r.country_region   as region,
                      p.prod_category,
                      sum(s.amount_sold) as amount_sold
                 from sales s join products  p    on p.prod_id    = s.prod_id
                              join customers cust on cust.cust_id = s.cust_id
                              join times     t    on t.time_id    = s.time_id
                              join countries r    on r.country_id = cust.country_id
                where calendar_year = 2000
                group by r.country_region, p.prod_category
                order by r.country_region, p.prod_category)
   loop
      if rec.region = 'Asia'
      then
         if rec.prod_category = 'Hardware'
         then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

Readable – Style 3 of ∞

trivadis

```
begin
   for rec in (
      select r.country_region as region,
             p.prod_category,
             sum(s.amount_sold) as amount_sold
        from sales s
        join products p
          on p.prod_id = s.prod_id
        join customers cust
          on cust.cust_id = s.cust_id
        join times t
          on t.time_id = s.time_id
        join countries r
          on r.country_id = cust.country_id
       where calendar_year = 2000
       group by r.country_region,
                p.prod_category
       order by r.country_region,
                p.prod_category
   ) loop
      if rec.region = 'Asia' then
         if rec.prod_category = 'Hardware' then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

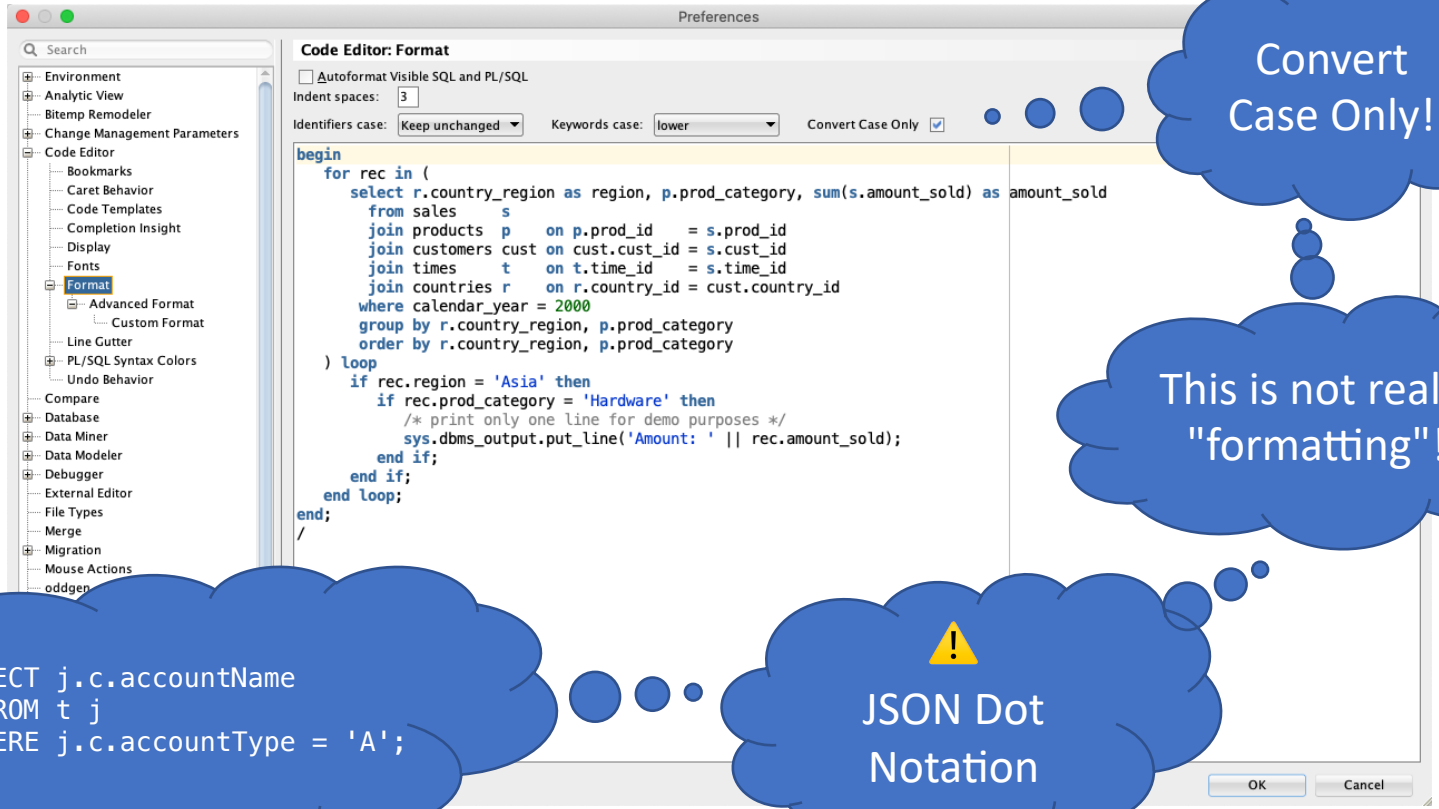
Readable – Style 4 of ∞

trivadis

```
begin
   for rec in (
      select r.country_region   as region,
             p.prod_category,
             sum(s.amount_sold) as amount_sold
        from sales      s
        join products   p
          on p.prod_id = s.prod_id
        join customers  cust
          on cust.cust_id = s.cust_id
        join times      t
          on t.time_id = s.time_id
        join countries  r
          on r.country_id = cust.country_id
       where calendar_year = 2000
       group by r.country_region,
                p.prod_category
       order by r.country_region,
                p.prod_category
   ) loop
      if rec.region = 'Asia' then
         if rec.prod_category = 'Hardware' then
            /* print only one line for demo purposes */
            sys.dbms_output.put_line('Amount: ' || rec.amount_sold);
         end if;
      end if;
   end loop;
end;
/
```

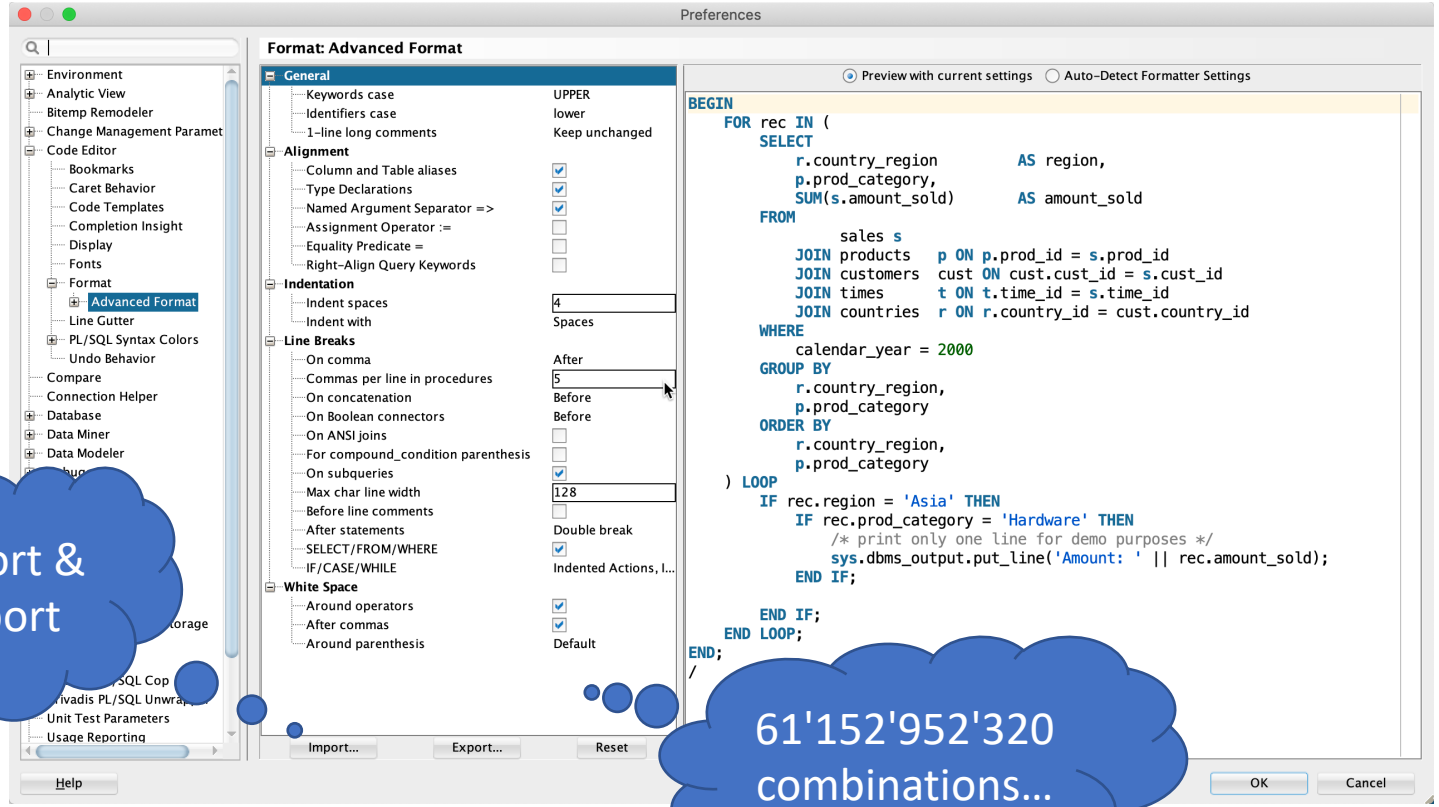
Format

trivadis



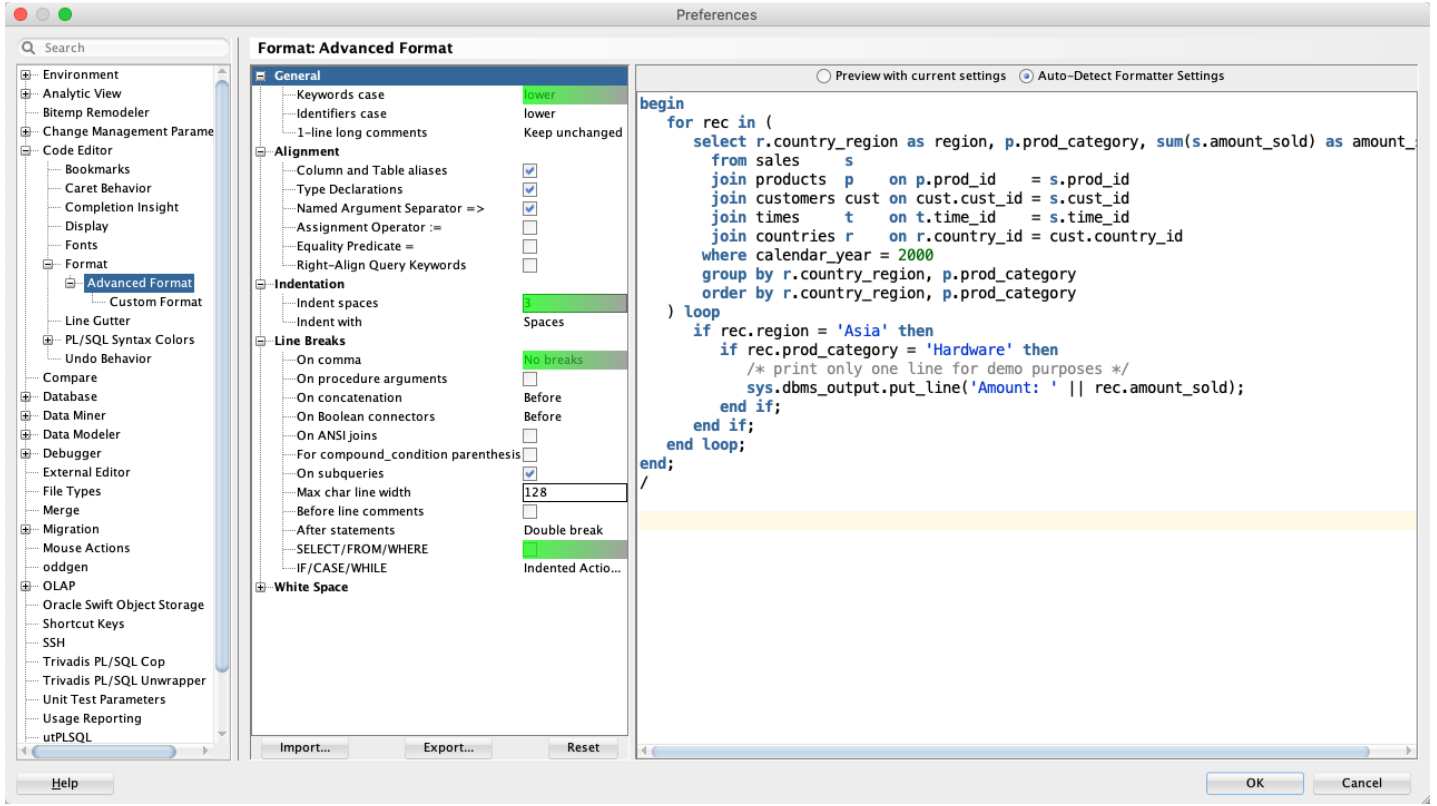
Advanced Format – Default Settings

trivadis

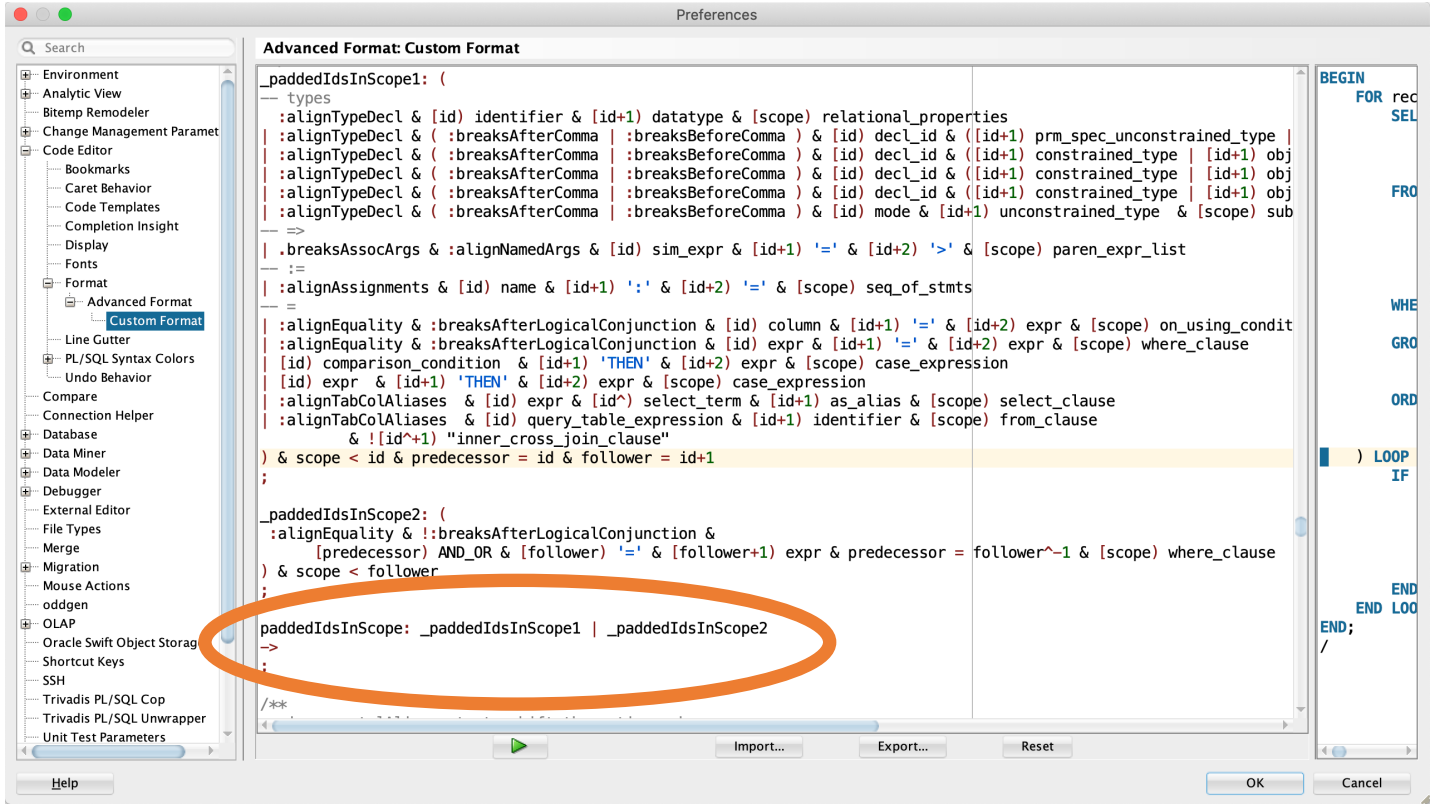


Auto-Detect Formatter Settings

trivadis



Custom Format

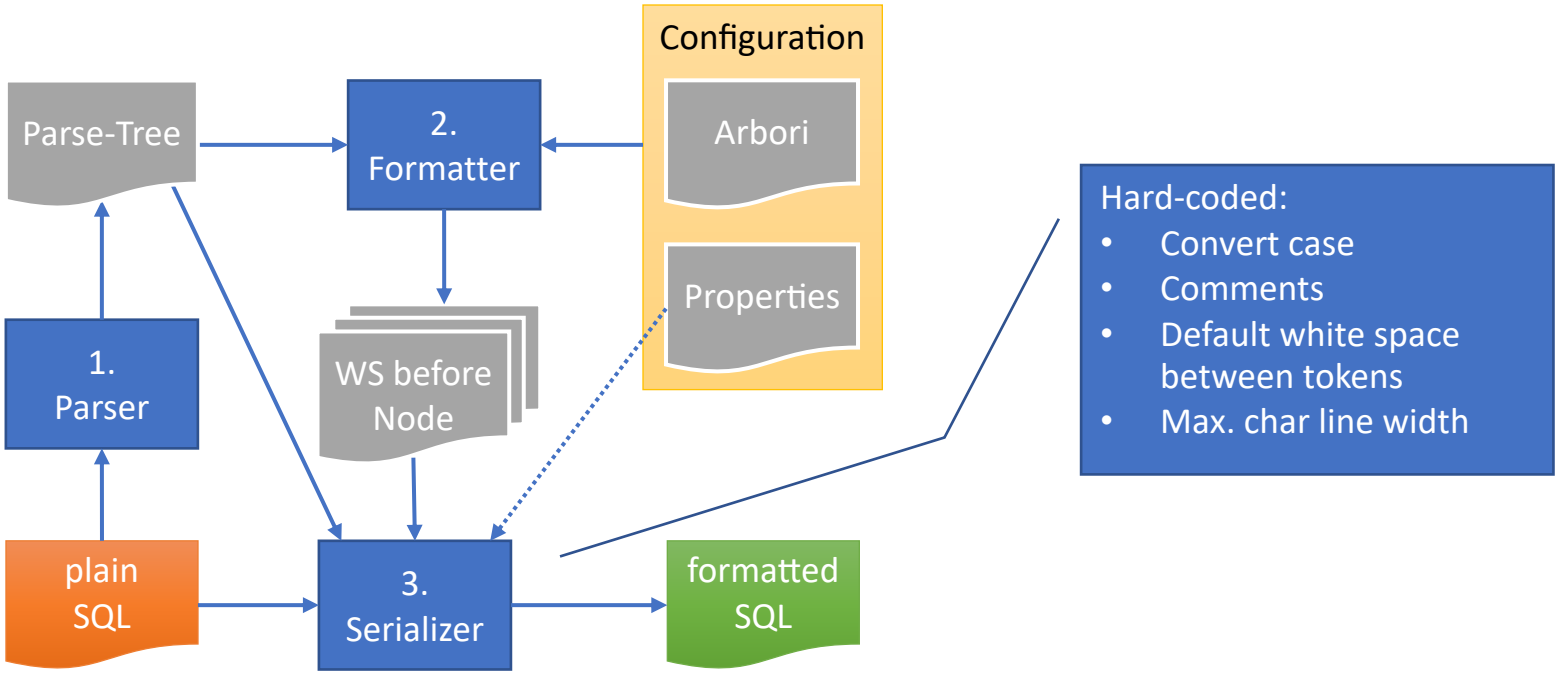


trivadis

Arbori – Part 2 (Formatter)

How the Formatter Works

trivadis



Java Callback Functions (->)

trivadis

```
public void … (Parsed target, Map<String, ParseNode> tuple) {…}
```

- indentedNodes1 (node)
- indentedNodes2 (node)
- skipWhiteSpaceBeforeNode (node)
- skipWhiteSpaceAfterNode (node)
- identifiers (identifier)
- extraBrkBefore (node)
- extraBrkAfter (node)
- brkX2 (node)
- rightAlignments (node)
- paddedIdsInScope (scope, predecessor, follower)
- incrementalAlignments (node)
- pairwiseAlignments (node, predecessor)
- ignoreLineBreaksBeforeNode (node)
- ignoreLineBreaksAfterNode (node)
- dontFormatNode (node)

See: https://www.salvis.com/blog/2020/11/12/formatter-callback-functions/

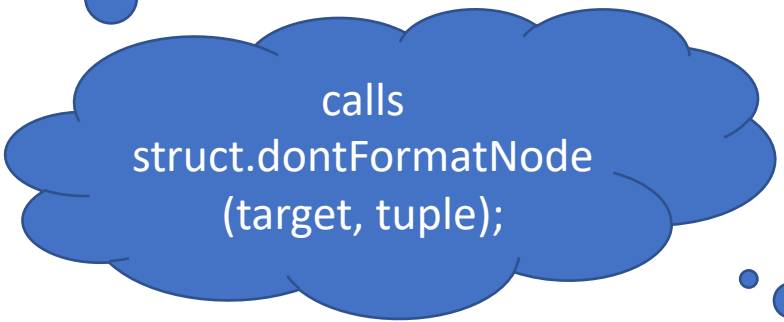
trivadis

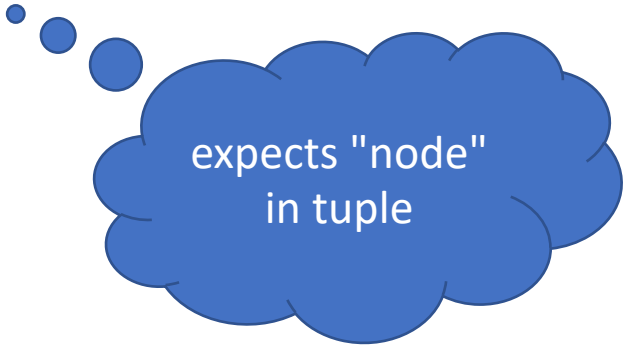
Demo 2

Java Callback Function Call – Example

trivadis

```
dontFormatNode:
   [node) numeric_literal | [node) select_term
   ->
;
```

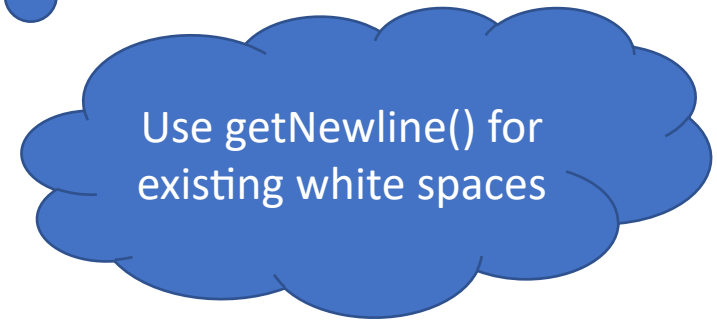




JavaScript Callback Function – Example

trivadis

```
aliasFix:
  [node) as_alias
  -> {
    var node = tuple.get("node");
    struct.putNewline(node.from, " ");
  }
;
```



Java Callback Function Call – Example

trivadis

```
_paddedIdsInScope1: (
…
-- fix: do not align as_alias for multiple select_term on the same line
| (:breaksBeforeComma | :breaksAfterComma) & :alignTabColAliases
            & [id) expr & [id^) select_term & [id+1) as_alias & [scope) select_clause
…
) & scope < id & predecessor = id & follower = id+1
;

paddedIdsInScope: _paddedIdsInScope1 | _paddedIdsInScope2
->
;
```

calls
struct.paddedIdsInScope
(target, tuple);

expects "scope",
"predecessor",
"follower" in tuple

trivadis

SQLcl

Built-in Format Command

trivadis

help format

```
FORMAT
---------

FORMAT BUFFER - formats the script in the SQLcl Buffer
FORMAT RULES <filename> - Loads SQLDeveloper Formatter rules file to formatter.
FORMAT FILE <input_file> <output_file>

Format used is default or for SQLcl can be chosen by setting an environmental variable
pointing to a SQLDeveloper export (.xml) of formatter options.
The variable is called SQLFORMATPATH
In SQLDeveloper the format options are the default chosen in the preferences.
```

Custom Formatter Command / JavaScript

trivadis

tvdformat

```
missing mandatory <rootPath> argument.

usage: tvdformat <rootPath> [options]

mandatory arguments:
  <rootPath>       file or path to directory containing files to format (content will be replaced!)
                   use * to format the SQLcl buffer

options:
  ext=<ext>        comma separated list of file extensions to process, e.g. ext=sql,pks,pkb
  mext=<ext>       comma separated list of markdown file extensions to process, e.g. ext=md,mdown
  xml=<file>       path to the file containing the xml file for advanced format settings
                   xml=default uses default advanced settings included in sqlcl
                   xml=embedded uses advanced settings defined in format.js
  arbori=<file>    path to the file containing the Arbori program for custom format settings
                   arbori=default uses default Arbori program included in sqlcl
```

See: https://github.com/Trivadis/plsql-formatter-settings/tree/main/sqlcl

trivadis

Core Messages

With Arbori You Can

trivadis

- Arbori is what makes highly customized code highlighting and formatting possible.
- Code Outline and the Arbori editor are helpful tools.
- Want to learn more?
  - https://vadimtropashko.wordpress.com/
  - https://www.salvis.com/blog/tag/arbori/
  - https://github.com/Trivadis/plsql-formatter-settings

You are welcome to join us at the Expo area.
We're looking forward to meeting you.

Link to the Expo area:
https://www.vinivia-event-manager.io/e/DOAG/portal/expo/29731

trivadis

trivadis