

OpenWorld 2008 UnConference

PL/SQL Conditional Compilation

Marcel Kratochvil

Piction

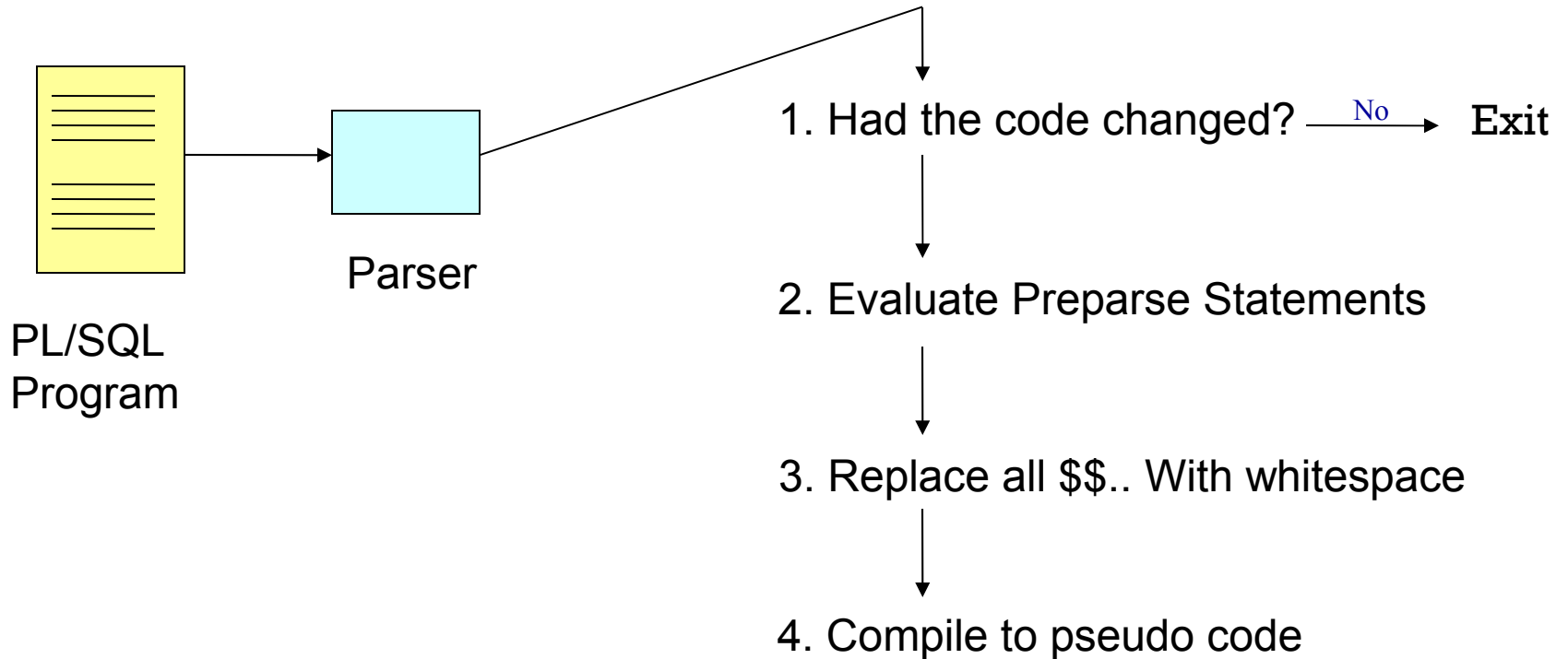
V1 (updated 1 Sept 2008)

Names

- Pre Parser
- Precompiler
- Conditional Compilation
- Macro

http://www.oracle.com/technology/tech/pl_sql/pdf/Plsql_Conditional_Compilation.pdf

Architecture



Pre Parser - History

- 9.2.0.6 (undocumented init.ora parameter)
- 10.1.0.4
- 10.2.0.1 - 10.2.0.4
- 11.1.0.6
- Conditional Compilation
 - Embed Debug/Tracing
 - Version Specific Constructs
 - Prototype alternate constructs
 - Component based installs

When to use it? (Official Line)

- Debug
 - `$if $$tracing $then`
- Assertions
 - `$if $$Asserting $then`
- Unit Testing

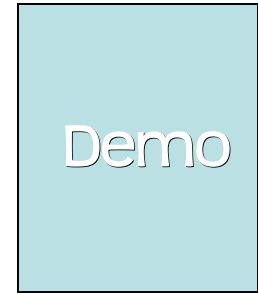
```
begin
  $if $$Testing $then
    mynewprog()
  $else
    raise Program_Error;
  $end
```

Real Life Scenarios Usage

- Oracle XE and Enterprise Edition
- Security
 - DMZ version
- Control features used (Oracle versions)

Syntax

- `$IF ... $THEN ...`
`$ELSIF ... $ELSE ...`
`$END`
- Alter session set
`Plsql_CCFlags = 'xxx:yyy';`
- `$ERROR ... $END`
- `$$TRACE_LEVEL`
- `$IF not(DECS.IS_DMZ) or`
`DECS.IS_DMZ is null $THEN`



Advanced Examples

```
$if $$Tracing $then  
  procedure debug(p1 in varchar2, p2 in varchar2)  
  begin  
    .. Advanced debug display commands  
  $end
```

....

```
$if $$Tracing  
  $then debug('in proc x','extra details yyy');  
$end
```



Expose Private procedures

```
Create package xxx  
as  
$if $$Tracing $then  
  procedure is_sensitive(....);  
$end
```

```
create package body xxx  
as  
  
$if not $$Tracing $then  
  procedure is_sensitive(....);  
$end
```



What If ?

Create procedure myprototype
as

```
$if $$Tracing $then  
  myvar varchar2(100);  
$else  
  myvar varchar2(10);  
$end
```

Begin

...

End;



- **Mock Object**

- Hard code return values for methods

- `$if $$mock_enabled $then`

- `return('fixed value');`

- `$else`

- `return(myvar);`

- `$end`

- Compare Different Structures

- ```
$if $$alt = 1 $then
 x integer;
$elsif $$alt = 2 $then
 x pls_integer;
$end
```

# Database Version Checking

- \$if DBMS\_Db\_Version.Ver\_LE\_9\_2 \$then

| Name        | Type        | Value | Description                    |
|-------------|-------------|-------|--------------------------------|
| VERSION     | PLS_INTEGER | 10    | Current version                |
| RELEASE     | PLS_INTEGER | 2     | Current release                |
| VER_LE_9    | BOOLEAN     | FALSE | Version <= 9                   |
| VER_LE_9_1  | BOOLEAN     | FALSE | Version <= 9 and release <= 1  |
| VER_LE_9_2  | BOOLEAN     | FALSE | Version <= 9 and release <= 2  |
| VER_LE_10   | BOOLEAN     | TRUE  | Version <= 10                  |
| VER_LE_10_1 | BOOLEAN     | FALSE | Version <= 10 and release <= 1 |
| VER_LE_10_2 | BOOLEAN     | TRUE  | Version <=10 and release <= 2  |

```
SQL> exec dbms_output.put_line(dbms_db_version.release);
2
```

```
SQL> exec dbms_output.put_line(dbms_db_version.version);
10
```



---

- Vendor Modules

- Component based installations
- Install all code
- Only run the code allowed

- Multi database version

## Support

```
$if DBMS_Db_Version.Ver_LE_10_1 $then
 return(TRUE);
$else
 image := ordsys.ordimage.init();
 image.source.localdata := src;
 metav := image.getMetadata('ALL');
$end
```

# That's It

---

- Conditional Compilation
  - Powerful
  - Extends PL/SQL
  - Just nice

Marcel Kratochvil  
Piction CTO  
[marcel@piction.com](mailto:marcel@piction.com)  
(m) +61416 093242  
<http://www.piction.com>  
Australia