Contents

[Introduction 2](#_Toc282170882)

[Package 2](#_Toc282170883)

[BSP application 2](#_Toc282170884)

[View 3](#_Toc282170885)

[Page attributes 5](#_Toc282170886)

[Layout 5](#_Toc282170887)

[Controller 5](#_Toc282170888)

[Class 6](#_Toc282170889)

[Attributes 8](#_Toc282170890)

[Internal types 10](#_Toc282170891)

[Methods 12](#_Toc282170892)

[Overloaded methods 12](#_Toc282170893)

[DO\_REQUEST 12](#_Toc282170894)

[Additionnal methods 14](#_Toc282170895)

[ZCPWR\_ANALYSE\_SCI\_DATA 15](#_Toc282170896)

[SEARCH\_FUNCTION 17](#_Toc282170897)

[SEARCH\_METHOD 18](#_Toc282170898)

[SEARCH\_PROG 18](#_Toc282170899)

[SEARCH\_INCLUDE 19](#_Toc282170900)

[Summary 20](#_Toc282170901)

[Test 21](#_Toc282170902)

[Request 21](#_Toc282170903)

[Response 21](#_Toc282170904)

# Introduction

This document describes the steps involved to implement the CAQS SAP bridge.

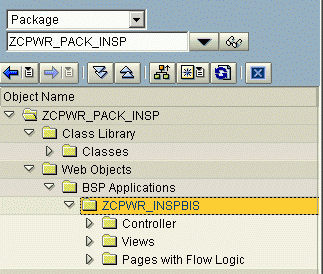
This bridge is used to remotely trigger SCI transaction from a client HTTP connexion and return code analysis results in a CSV (comma separated values) format.

# Package

Create a package named « ZCPWR\_PACK\_INSP ».

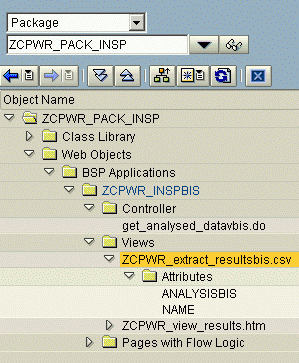
## BSP application

In this package, create a BSP application named “ZCPWR\_INSPBIS”.

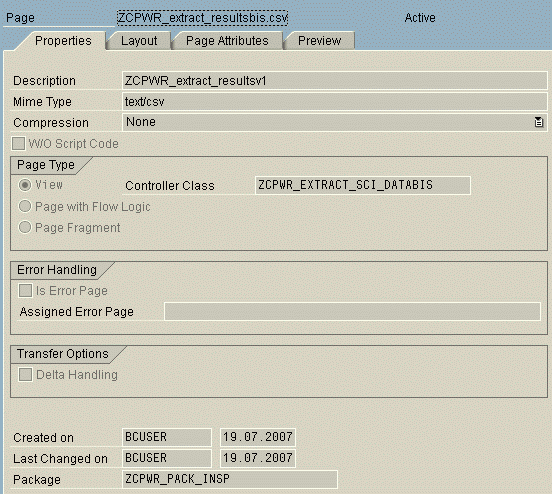


### View

In the ZCPWR\_INSPBIS BSP application, create a view and name it “ZCPWR\_extract\_resultsbis.csv”.



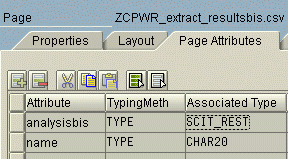
NB: Make sure it has “ZCPWR\_EXTRACT\_SCI\_DATABIS” as its controller class.



#### Page attributes

Add the following page attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **TypingMeth** | **Associated Type** |
| analysisbis | TYPE | SCIT\_REST |
| name | TYPE | CHAR20 |



#### Layout

Insert the following layout content:

<%@page language="abap"%>

<%

data ana\_line like line of analysisbis.

loop at analysisbis into ana\_line.

%>

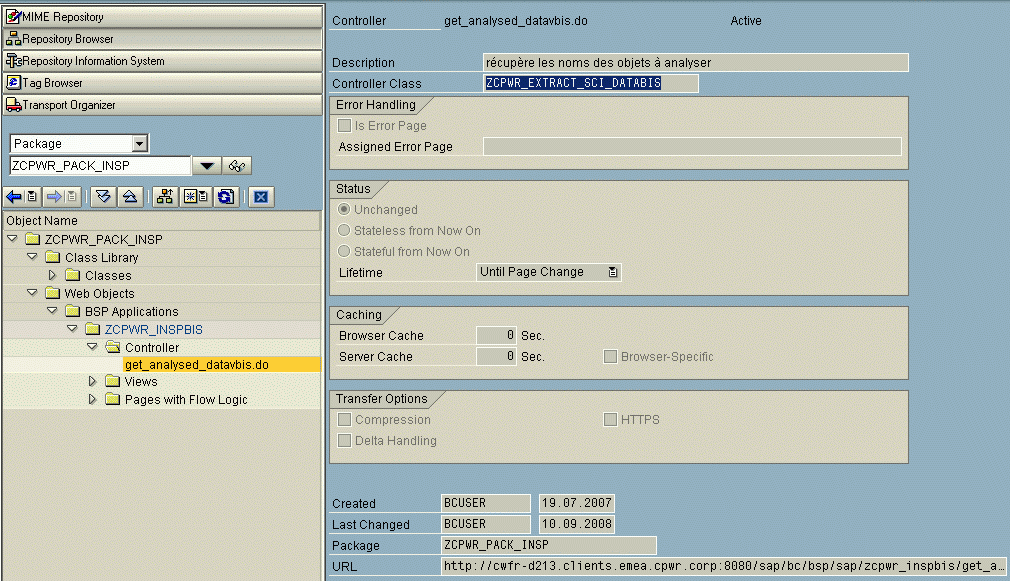
<%= ana\_line-test.%>;<%= ana\_line-objname.%>;<%= ana\_line-objtype.%>;<%= ana\_line-sobjname.%>;<%= ana\_line-sobjtype.%>;<%= ana\_line-PARAM1.%>;<%= ana\_line-line.%>;<%= ana\_line-kind.%><%endloop.%>

### Controller

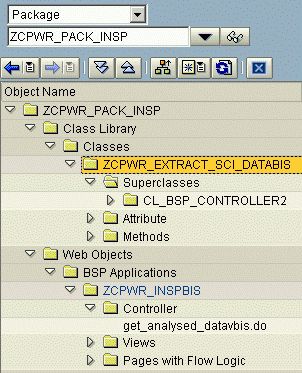
In this BSP, create a controller and name it « get\_analysed\_datavbis.do ».

## Class

Create the associated controller class as “ZCPWR\_EXTRACT\_SCI\_DATABIS”.



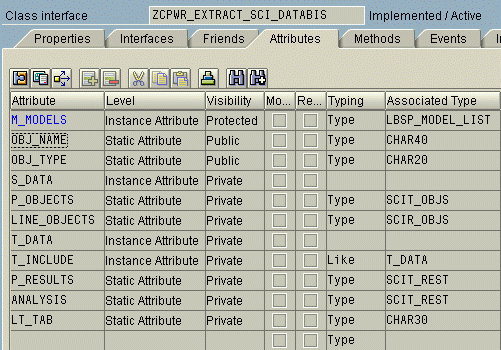
NB: this controller class must extend « CL\_BSP\_CONTROLLER2”.



### Attributes

Add the following attributes :

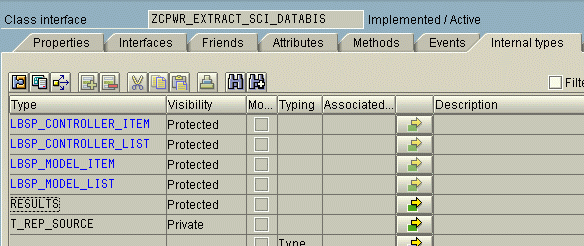
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Level | Visibility | Typing | Associated type |
| OBJ\_NAME | Static Attribute | Public | Type | CHAR40 |
| OBJ\_TYPE | Static Attribute | Public | Type | CHAR20 |
| S\_DATA | Instance Attribute | Private |  |  |
| P\_OBJECTS | Static Attribute | Private | Type | SCIT\_OBJS |
| LINE\_OBJECTS | Static Attribute | Private | Type | SCIR\_OBJS |
| T\_DATA | Instance Attribute | Private |  |  |
| T\_INCLUDE | Instance Attribute | Private | Like | T\_DATA |
| P\_RESULTS | Static Attribute | Private | Type | SCIT\_REST |
| ANALYSIS | Static Attribute | Private | Type | SCIT\_REST |
| LT\_TAB | Static Attribute | Private | Type | CHAR30 |



### Internal types

Add the following internal types :

|  |  |
| --- | --- |
| **Type** | **Visibility** |
| RESULTS | Protected |
| T\_REP\_SOURCE | Private |



Create the following definition for type “RESULTS”:

\*"\* protected components of class ZCPWR\_EXTRACT\_SCI\_DATA

\*"\* do not include other source files here!!!

protected section.

types:

BEGIN OF results,

object\_type TYPE char04,

object\_name TYPE string,

subobject\_type TYPE char04,

subobject\_name TYPE char40,

incl\_name TYPE char40,

END OF results .

Create the following definition for type “T\_REP\_SOURCE”:

\*"\* private components of class ZCPWR\_EXTRACT\_SCI\_DATA

\*"\* do not include other source files here!!!

private section.

types:

BEGIN OF t\_rep\_source,

line(10000),

END OF t\_rep\_source .

data:

BEGIN OF s\_data,

object\_type TYPE char04,

object\_name TYPE string,

subobject\_type TYPE char04,

subobject\_name TYPE char40,

incl\_name type char40,

END OF s\_data .

class-data P\_OBJECTS type SCIT\_OBJS .

class-data LINE\_OBJECTS type SCIR\_OBJS .

data:

t\_data LIKE TABLE OF s\_data .

type-pools T .

data T\_INCLUDE like T\_DATA .

class-data P\_RESULTS type SCIT\_REST .

class-data ANALYSIS type SCIT\_REST .

class-data LT\_TAB type CHAR30 .

methods SEARCH\_FUNCTION .

methods SEARCH\_METHOD .

methods SEARCH\_PROG .

methods SEARCH\_INCLUDE

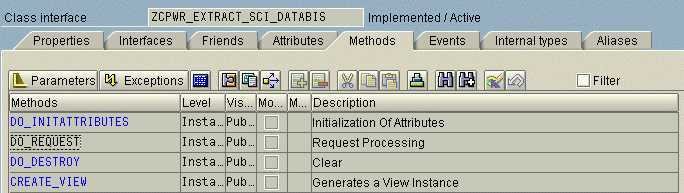
importing

!P\_LINE type T\_REP\_SOURCE optional .

### Methods

#### Overloaded methods

Select method named “DO\_REQUEST”,



and overload it with the following code :

##### DO\_REQUEST

METHOD DO\_REQUEST.

obj\_name = ''.

lt\_tab = request->get\_form\_field( 'pkg' ).

TYPES: BEGIN OF t\_rep\_source,

line(10000),

END OF t\_rep\_source.

DATA: l\_i\_prog\_source TYPE TABLE OF t\_rep\_source ,

l\_was TYPE t\_rep\_source.

DATA: "p\_objects TYPE scit\_objs,

line\_objects TYPE scir\_objs,

files TYPE TABLE OF tadir,

line LIKE LINE OF files.

TYPES: BEGIN OF tmethod,

cmpname TYPE char30,

descript TYPE char30,

exposure TYPE int4,

methodkey TYPE string,

END OF tmethod.

DATA : clsn TYPE seoclsname.

TYPES : BEGIN OF seop\_method\_w\_include,

cpdkey TYPE seocpdkey,

incname TYPE programm,

END OF seop\_method\_w\_include.

DATA methodline TYPE seop\_method\_w\_include.

DATA seop\_methods\_w\_include TYPE TABLE OF seop\_method\_w\_include .

DATA: myview TYPE REF TO if\_bsp\_page.

myview = create\_view( view\_name = 'ZCPWR\_extract\_resultsbis.csv' ).

SELECT \* FROM tadir INTO TABLE files.

LOOP AT files INTO line.

\* IF line-author EQ 'BCUSER'.

IF line-devclass EQ lt\_tab.

line\_objects-objtype = line-object.

line\_objects-objname = line-obj\_name.

line\_objects-responsibl = line-author.

line\_objects-devclass = 'DEVEL'.

line\_objects-prgname = ''.

INSERT line\_objects INTO TABLE p\_objects.

ENDIF.

ENDLOOP.

CALL METHOD search\_function.

CLEAR line\_objects.

CALL METHOD search\_method.

CLEAR line\_objects.

CALL METHOD search\_prog.

IF t\_data IS INITIAL.

ENDIF.

DATA previous TYPE char40.

previous = ''.

LOOP AT t\_data INTO s\_data.

IF s\_data-object\_type = 'FUGR' OR s\_data-object\_type = 'PROG'.

IF s\_data-incl\_name <> previous.

previous = s\_data-incl\_name.

obj\_name = s\_data-object\_name.

obj\_type = s\_data-object\_type.

REFRESH l\_i\_prog\_source .

READ REPORT s\_data-incl\_name INTO l\_i\_prog\_source .

CHECK sy-subrc = 0.

LOOP AT l\_i\_prog\_source INTO l\_was.

CALL METHOD search\_include

EXPORTING

p\_line = l\_was.

ENDLOOP.

ENDIF.

ENDIF.

ENDLOOP.

write( '<html><body>' ).

myview->set\_attribute( name = 'name' value = sy-uname ).

CALL METHOD zcpwr\_analyse\_sci\_data.

myview->set\_attribute( name = 'analysisbis' value = analysis ).

call\_view( myview ).

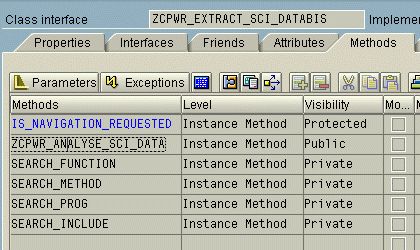
write( '</body></html>' ).

ENDMETHOD.

#### Additionnal methods

Create the following additionnal methods:

|  |  |  |
| --- | --- | --- |
| **Methods** | **Level** | **Visibility** |
| ZCPWR\_ANALYSE\_SCI\_DATA | Instance Method | Public |
| SEARCH\_FUNCTION | Instance Method | Private |
| SEARCH\_METHOD | Instance Method | Private |
| SEARCH\_PROG | Instance Method | Private |
| SEARCH\_INCLUDE | Instance Method | Private |



##### ZCPWR\_ANALYSE\_SCI\_DATA

Insert the following code for method “ZCPWR\_ANALYSE\_SCI\_DATA”:

METHOD ZCPWR\_ANALYSE\_SCI\_DATA .

DATA: p\_objects TYPE scit\_objs,

line\_objects TYPE scir\_objs,

s\_results TYPE scir\_rest,

files TYPE TABLE OF tadir,

line LIKE LINE OF files,

name(40),

nommeth(40).

TYPES: BEGIN OF tmethod,

cmpname TYPE char30,

descript TYPE char30,

exposure TYPE int4,

methodkey TYPE string,

END OF tmethod.

DATA: dumimethods TYPE STANDARD TABLE OF tmethod.

DATA:ilocmethods LIKE dumimethods[].

DATA: locmethodkey TYPE program.

DATA: methodid TYPE seocpdkey.

DATA: imethods TYPE STANDARD TABLE OF tmethod.

DATA: classname TYPE char30.

DATA : amethod TYPE tmethod.

LOOP AT t\_data INTO s\_data.

line\_objects-objtype = s\_data-object\_type.

line\_objects-objname = s\_data-object\_name.

line\_objects-responsibl = ''.

line\_objects-devclass = ''.

line\_objects-prgname = ''.

INSERT line\_objects INTO TABLE p\_objects.

ENDLOOP.

CALL FUNCTION 'SCI\_INSPECT\_LIST'

EXPORTING

p\_user = sy-uname

p\_objects = p\_objects

p\_name = 'DEFAULT'

IMPORTING

p\_results = p\_results

EXCEPTIONS

no\_object = 1

objs\_already\_exists = 2

no\_default\_checkvariant = 3

too\_many\_objects = 4

could\_not\_read\_variant = 5

OTHERS = 6.

IF p\_results IS INITIAL.

ENDIF.

DATA : clsn TYPE seoclsname.

TYPES : BEGIN OF seop\_method\_w\_include,

cpdkey TYPE seocpdkey,

incname TYPE programm,

END OF seop\_method\_w\_include.

DATA methodline TYPE seop\_method\_w\_include.

DATA seop\_methods\_w\_include TYPE TABLE OF seop\_method\_w\_include .

LOOP AT p\_results INTO s\_results.

DATA rulename TYPE char30.

IF s\_results-objname <> s\_results-sobjname.

LOOP AT t\_data INTO s\_data.

IF s\_results-objname = s\_data-object\_name

AND s\_results-sobjname = s\_data-subobject\_name.

write( 'ok : ' ).write( s\_results-objname ).write( s\_results-sobjname ). write( '<br>' ).

ENDIF.

ENDLOOP.

ENDIF.

IF NOT s\_results-code CS 'MESSAGEG'.

CONCATENATE s\_results-test '\_' s\_results-code INTO name.

nommeth = ''.

IF s\_results-objtype EQ 'CLAS'.

clsn = s\_results-objname.

nommeth = s\_results-sobjname.

LOOP AT t\_data INTO s\_data.

IF s\_data-object\_name = s\_results-objname AND s\_data-incl\_name = s\_results-sobjname.

nommeth = s\_data-subobject\_name.

EXIT.

ENDIF.

ENDLOOP.

ENDIF.

s\_results-test = name.

s\_results-param1 = nommeth.

APPEND s\_results TO analysis.

ENDIF.

ENDLOOP.

DATA found TYPE c.

found = ''.

LOOP AT t\_data INTO s\_data.

found = ''.

LOOP AT analysis INTO s\_results.

IF s\_data-object\_name EQ s\_results-objname.

found = 'X'.

ENDIF.

ENDLOOP.

IF found EQ ''.

CLEAR s\_results.

s\_results-objname = s\_data-object\_name.

s\_results-objtype = s\_data-object\_type.

s\_results-sobjname = s\_data-subobject\_name.

s\_results-sobjtype = s\_data-subobject\_type.

APPEND s\_results TO analysis.

ENDIF.

ENDLOOP.

ENDMETHOD.

##### SEARCH\_FUNCTION

Insert the following code for method “SEARCH\_FUNCTION”:

method SEARCH\_FUNCTION .

\*&--------------------------------------------------------------------\*

\*& Form search\_function

\*&--------------------------------------------------------------------\*

data :

begin of enfline,

FUNCNAME type char30,

AREA type char26,

ACTIVE type char1,

GENERATED type char1,

FREEDATE type char8,

GLOBAL type char1,

LOC\_PRIV type char1,

EXTEN1 type char1,

EXTEN2 type char1,

EXTEN3 type char1,

EXTEN4 type char1,

EXTEN5 type char1,

end of enfline.

data : begin of tfline,

FUNCNAME type char30,

PNAME type char40,

INCLUDE type char2,

FREEDATE type char8,

APPL type char1,

MAND type char3,

FMODE type char1,

HOST type char8,

UTASK type char1,

end of tfline.

TYPES: BEGIN OF t\_rep\_source,

line(10000),

END OF t\_rep\_source.

DATA: l\_i\_prog\_source TYPE TABLE OF t\_rep\_source ,

l\_was TYPE t\_rep\_source.

LOOP AT p\_objects INTO line\_objects.

IF line\_objects-objtype = 'FUGR'.

SELECT \* FROM enlfdir into enfline WHERE area = line\_objects-objname.

SELECT \* FROM tfdir into tfline WHERE funcname = enfline-funcname.

"pname : nom à recuperer pour les includes

s\_data-object\_type = 'FUGR'.

s\_data-object\_name = enfline-area.

s\_data-subobject\_type = 'FUNC'.

s\_data-subobject\_name = tfline-funcname.

s\_data-incl\_name = tfline-pname.

APPEND s\_data TO t\_data.

ENDSELECT.

ENDSELECT.

ENDIF.

ENDLOOP.

endmethod.

##### SEARCH\_METHOD

Insert the following code for method “SEARCH\_METHOD”:

method SEARCH\_METHOD .

DATA : clsn TYPE seoclsname.

TYPES : BEGIN OF seop\_method\_w\_include,

cpdkey TYPE seocpdkey,

incname TYPE programm,

END OF seop\_method\_w\_include.

DATA methodline TYPE seop\_method\_w\_include.

DATA seop\_methods\_w\_include TYPE TABLE OF seop\_method\_w\_include .

LOOP AT p\_objects INTO line\_objects.

IF line\_objects-objtype EQ 'CLAS'.

clsn = line\_objects-objname.

cl\_oo\_classname\_service=>get\_all\_method\_includes(

EXPORTING clsname = clsn

RECEIVING result = seop\_methods\_w\_include

EXCEPTIONS

class\_not\_existing = 2 ).

LOOP AT seop\_methods\_w\_include INTO methodline.

s\_data-object\_type = line\_objects-objtype.

s\_data-object\_name = clsn.

s\_data-subobject\_type = 'METH'.

s\_data-subobject\_name = methodline-cpdkey-cpdname.

s\_data-incl\_name = methodline-incname.

APPEND s\_data TO t\_data.

ENDLOOP.

ENDIF.

ENDLOOP.

endmethod.

##### SEARCH\_PROG

Insert the following code for method “SEARCH\_PROG”:

METHOD SEARCH\_PROG .

LOOP AT p\_objects INTO line\_objects.

IF line\_objects-objtype = 'PROG'.

s\_data-object\_type = line\_objects-objtype.

s\_data-object\_name = line\_objects-objname.

s\_data-subobject\_type = ''.

s\_data-subobject\_name = ''.

s\_data-incl\_name = line\_objects-objname.

APPEND s\_data TO t\_data.

ENDIF.

ENDLOOP.

ENDMETHOD.

##### SEARCH\_INCLUDE

Insert the following code for method “SEARCH\_INCLUDE”:

METHOD SEARCH\_INCLUDE .

TYPES:

BEGIN OF t\_program,

prog TYPE trdir-name,

subc TYPE trdir-subc,

text TYPE trdirt-text,

sel(1),

END OF t\_program.

DATA:

l\_pozicia TYPE i,

l\_wa\_incl\_search(72),

l\_include(72),

l\_next\_search TYPE t\_rep\_source.

DATA: g\_i\_progs TYPE STANDARD TABLE OF t\_program.

DATA t\_prog\_line TYPE t\_program.

IF p\_line-line CP '\*INCLUDE\*'.

CLEAR: l\_include, l\_next\_search.

MOVE sy-fdpos TO l\_pozicia.

\* Include found. Check if it is not a comment or something like that.

CHECK NOT p\_line-line+0(1) = '\*' AND

NOT p\_line-line+0(1) = '"' .

SEARCH p\_line-line FOR '"' STARTING AT 1 ENDING AT l\_pozicia.

CHECK sy-subrc <> 0.

ADD 7 TO l\_pozicia.

MOVE p\_line-line+l\_pozicia TO l\_wa\_incl\_search.

SHIFT l\_wa\_incl\_search LEFT DELETING LEADING space.

CHECK NOT l\_wa\_incl\_search+0(1) = '"'.

CHECK l\_wa\_incl\_search NP 'STRUCTURE\*' AND

l\_wa\_incl\_search NP 'TYPE\*'.

IF l\_wa\_incl\_search CA '.'.

l\_pozicia = sy-fdpos.

CHECK l\_pozicia > 0.

MOVE l\_wa\_incl\_search+0(l\_pozicia) TO l\_include.

CONDENSE l\_include NO-GAPS.

MOVE l\_wa\_incl\_search+l\_pozicia TO l\_next\_search-line.

ELSE.

CONDENSE l\_wa\_incl\_search NO-GAPS.

MOVE l\_wa\_incl\_search TO l\_include.

ENDIF.

CHECK NOT l\_include IS INITIAL.

TRANSLATE l\_include TO UPPER CASE.

READ TABLE g\_i\_progs INTO t\_prog\_line WITH KEY prog = l\_include sel = 'X' .

IF sy-subrc <> 0.

CLEAR g\_i\_progs.

SELECT SINGLE trdir~name trdir~subc trdirt~text

INTO t\_prog\_line

FROM trdir LEFT JOIN trdirt

ON trdir~name = trdirt~name

AND trdirt~sprsl = sy-langu

WHERE trdir~name = l\_include .

data sline like line of t\_include.

IF sy-subrc = 0.

sline-object\_type = obj\_type.

sline-object\_name = obj\_name.

sline-subobject\_type = 'INCL'.

sline-subobject\_name = l\_include.

sline-incl\_name = ''.

APPEND sline TO t\_include.

ENDIF.

ENDIF.

IF NOT l\_next\_search IS INITIAL.

CALL METHOD me->search\_include

EXPORTING

p\_line = l\_next\_search.

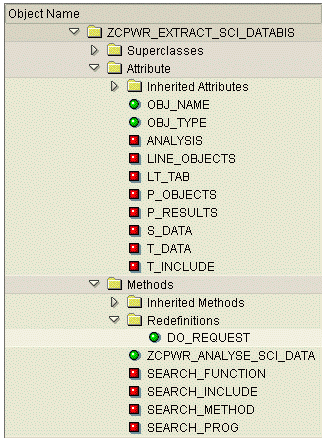
ENDIF.

ENDIF.

ENDMETHOD.

### Summary

NB: you should end up with the following class structure:



# Test

Check and activate all parts.

## Request

From a Web browser, test the URL :

http://<server>:<port>/sap/bc/bsp/sap/zcpwr\_inspbis/get\_analysed\_datavbis.do?pkg=<packageName>

NB: replace <server> with the actual name of your SAP Web server.

NB: replace <port> with the actual port (defaults to 80).

NB: replace <packageName> with the actual name of a development package containing ABAP objects to analyse with Code Inspector.

## Response

The response should be formatted like the following example :

CL\_CI\_TABNAMES\_PUBLIC\_0001;ZCPWR\_EXTRACT\_SCI\_DATABIS;CLAS;ZCPWR\_EXTRACT\_SCI\_DATABIS=====CM001;PROG;SEARCH\_FUNCTION;000042;N

CL\_CI\_TABNAMES\_PUBLIC\_0001;ZCPWR\_EXTRACT\_SCI\_DATABIS;CLAS;ZCPWR\_EXTRACT\_SCI\_DATABIS=====CM001;PROG;SEARCH\_FUNCTION;000043;N

CL\_CI\_TABNAMES\_PUBLIC\_0001;ZCPWR\_EXTRACT\_SCI\_DATABIS;CLAS;ZCPWR\_EXTRACT\_SCI\_DATABIS=====CM002;PROG;SEARCH\_INCLUDE;000050;N

CL\_CI\_TABNAMES\_PUBLIC\_0001;ZCPWR\_EXTRACT\_SCI\_DATABIS;CLAS;ZCPWR\_EXTRACT\_SCI\_DATABIS=====CM002;PROG;SEARCH\_INCLUDE;000050;N

*[…]*