COCURRENT PROGRAM CONCEPTS

1. **Delete concurrent program**
   a. We cannot delete concurrent program directly from the from end application. Instead we can disable it. We can delete it from the back end by using API

   ```
   Begin
   fnd_program.delete_program('shortname','applicatoin_name');
   End;

   Commit;
   ```

   In the same way we can delete executables, request group, parameters, and incompatibilities.

   We can use fnd_program.program_exists to check the existence of program.

2. **Value sets**
   a. With the value sets we can display the list of values for parameter. It is similar is item clause in discoverer.
   b. There are 8 limited value sets
      i. Table
      ii. Dependent
      iii. Independent
      iv. None
      v. Special
      vi. Pair
      vii. Translatable dependent
      viii. Translatable independent
   c. We can create value sets in application developer or system administrator.

a. **Table value sets:**

Goto application developer➔application➔validation➔set
Enter the name, format, width
Select the validation type as table
Click on edit information
Provide the table po_vendors
Application name—payables
Enter the column name as vendor_name
Datatype, width, click on save.

--goto concurrent program→query with your request→click on parameter→attach the value set→click on save.
--switch responsibility to your responsibility (payable)
--submit your report. It will popup the parameter window.
--click on LOV. It will show the list of values.

➡ Avoid duplicate values in the table value set:

First method:

Use distinct keyword in the select statement and mention the entire statement In the table name place.

EX: (select distinct segment1 form po_headers_all) a under the table name
a.segment1----under the column name

Second method:

Create a view on top of distincted statement like below in the custom schema
Create or replace view hbo_po_list_v as
Select distinct segment1 from po_headers_all

Create synonym name like below:
Create public synonym hbo_po_list_v for apps.hbo_po_list_v

Provide grand access to synonym
Grant all on hbo_po_list_v

Use this view in creating the table value set.

➡ Dynamic relation b/w two table value sets:

There are two table value sets one is for suppliers and other is for supplier sites.
If we select a particular supplier in the first parameter, related supplier sites we need to see in the second parameter.
For this we use $FLEX$. This is case sensitive.

Approach

Create one table value set. Take the column name, table name and its id value. That is common join “vendor_id”. Take other value set in the where clause give the below condition
Vendor_id=:$FLEX$.previous_valveset_name
Eg: first value set name: hbo_sop
Table name: po_vendors

Column_name: vendor name
Id: vendor_id

Second value set name: hbo_site
Table name: po_vendor_sites_all
Column name: vendor_site_code
Id: vendor_site_id
Where: where vendor_id=:$FLEX$.hbo_sop.

➡ **Additional columns in table value sets:**

It displaying customer number along with this we can additional values

b. NONE VALUE SETS:

With none value set we can put restriction like below
*to pass numbers only
* to pass value with in the range (b/w 1000 and 8000).

Navigation:

Application developer➔validation➔set
Select the validation type as none
Enter the min value and max value
Finally save.

Goto concurrent program➔define
Query with your concurrent program
Goto parameter attach it
Save.

Goto your responsibility➔submit your report

If we enter character in the parameter it will give the fallowing error
“APC-FND-0126: valid characters are 10-100”

If we enter values out of range you will get the fallowing error.
“please enter b/w xx and xxx”

c. INDEPENDENT VALUE SET:

In this value set we can enter the values manually
Navigation:

Application developer➔application➔validation➔set
Enter value set name
Choose the validation type as independent set

Goto application➔validation➔value
Enter the value set name click on find
Enter the values manually and save.

d. DEPENDENT VALUE SET:

Dependent value set is depends always on one independent value set. To first parameter we must attach independent value set. For second parameter we can attach dependent value set. It is very similar to $FLEX$ behavior but we need to provide the values manually instead of providing from data base.

Approach;
First we create one independent value set. Provide the list values like below.
Ge plastics
Advantage corp

Goto application➔validation➔set
Enter the value set name
Choose the validation type as dependent.
Click on edit information attach the independent value set.
Provide the default valve from the first value set and its description.

Goto application➔validation➔values
Enter the dependent value set➔find➔enter the list of values for the first value of independent. To provide the list of values to the second value. Press down arrow likewise we can provide the value for all the independent value set values.

Attach the value set to the second parameter.
Save.
Submit it from your responsibility.

e. DEFAULT TYPES:

With the default types we can provide the default value to the parameters at srs window (standard request submission) one more significance is we can avoid hardcoded values into the query part.

There are 6 default types:
---constant
---profile
---sql statement
---current date
---current time
---segment

1. **Constant: we need to enter the values manually in the concurrent program windows. No value set is associated with it.**

2. **Profile:**

   To pick up the values from the apps environment
   Eg: report_name: 1099 payment report
   Parameter: sob (set of books)
   Default type: profile
   Value: gl_set_of_bks_id.

   Here sob values will vary for every responsibility.
   Select set_of_books, name from gl_set_of_books

   1. **Vision operation**
   2. **Vision distributions.**

   To define profile. Goto system administrator→profile→system
   We can define profile options at various levels site, application, responsibility, user.

   Even priority also in the same order site= lowest.user

   To find out the profile values, whether enabled or not at user/site levels etc.
   Use the query(check email).

3.sql statement:

We can refer the look up values through the select statement to parameters
Eg:
Report name: 1099 payment reports

Parameter: reporting method
Default type: sql statements
Default value: select displayed_field from ap_lookup_code where lookup_type='group_by' and lookup_code='vendor'

For every module we can see one lookup table
Ar_ar_lookups
Ap_ap_lookup_codes
Pa_pa_lookups

Navigation: payables vison operatio➔setups➔lookups➔payables

In application developer also we can see this details: common utility parameter
Eg: yes-no parameter
Nav: application developer➔application➔lookups➔common

4.current date:
We can provide system date to the parameters
Report name: 1099 payment report
Parameter: to payment date
Default type: current date
Default value: _____

5. current time:
We can pass date along with time.

6.segment:
The first parameter be copied to next parameter

This functionality we can achieve by using the sql statement type like below
Default type: sql statement
Value: select :$flex$.bw_sop from dual.

If there is a space in previous parameter name we cannot use this. It will give the issue
Like invalid reference XXX in default value defined for segment second parameter
Name.

For date:
Value set: fnd_standard_date

Default value:

```
        Select fnd_date.date,
to_chardate(fnd_date.canonical_to_date(:$flex$.fnd_standard_date)) from dual;
```

Trace file:

the main use of enabling trace for a concurrent program comes during performance tuning .by examining a trace file, we come to know which