COBOL DB2 rules configuration
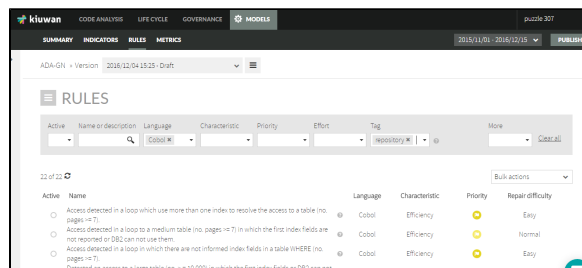
How to configure COBOL DB2 rules

COBOL technology has a group of rules, known as COBOL DB2 rules, that provide quite accurate insights on performance and efficiency issues related to DB2 access.

They have been proved to find performance issues and to provide remediation clues.

You can find COBOL DB2 rules in Models Management - Rules, and filtering by **language** = **cobol** and **tag** = **repository**.

Configuration of COBOL DB2 rules

To properly work, COBOL DB2 rules need to be fed with DB2 information.

All together, the DB2 information and the rules' logic provide quite a powerful performance issues detection system.

Their rules consume DB2 information from two specific resource files (**catalog** and **plantable** files).

These resource files are tightly coupled to your DB2 installation so you should provide them so Kiuwan can apply its efficiency logic.

Below you can find the way to specify location and format of these files.

Regarding format, every file follows a specific (fixed) sequence of information fields with a default size.

> ⓘ You can modify default sizes but keep always the order, as the rules will read those files according to the specified sequence.

DB2 Catalog file

This file contains information related to tables being used by the program to be analyzed. In case the program does not access DB2, this file will be empty.

Information contained in this file is organized by table and for every table is divided into three sections:

- Table Line format (one record per table, 37 characters)
- Indexes Line format (one record per index and table, 20 characters)

- Index Fields Line format (one record per field of the index, 24 characters, and records ordered according to the order that appears in the index)

There are two properties defined in ${kiuwan_local_analyzer_installation_dir}/conf/analyzer.properties file to configure DB2 Catalog file:

1. cobol.db2.catalog.path
2. cobol.db2.catalog.format

cobol.db2.catalog.path

This property defines where the DB2 catalog file is located.

After the 'file:///' prefix, encode the path where to find the resource files.

You can define an absolute path or a relative one. Use the '@{src.dir}' pattern to set a path relative to the source directory of the analysis.

```
cobol.db2.catalog.path=file:///@{src.dir}/resources/catalog.txt
```

cobol.db2.catalog.format

This property is used to define the inner format of Catalog file.

```
# Format definition for Catalog resource.
cobol.db2.catalog.format=2:18_20:9_29:9_2:18_20:(1)_21:9_30:9_2:18_20:5
```

Catalog file example:


```
cobol.db2.catalog.format = 2:18_20:9_29:9_2:18_20:(1)_21:9_30:9_2:18_20:5
```

NAME	TYPE	POSITION default	LENGTH default		DESCRIPTION
T	unmodifiable*	1	1		Table description
TNAME	alphanumeric	2	18		Table name
TCARD	numeric	20	9		Table cardinality
TNPAG	numeric	29	9		Table number of pages
I	unmodifiable*	1	1		Index desctiption
INAME	alphanumeric	2	18		Index name
ITYPE	char	20	(1)		Index type: U(nic), D(uplicate), ...
IFIRSTKEYCARD	numeric	21	9		Index first key cardinality
ICARD	numeric	30	9		Index cardinatlity
C	unmodifiable*	1	1		Column description
CNAME	alphanumeric	2	18		Column name
COLNO	numeric	20	5		Field order in the table

(*) T, I and C characters are unmodifiable, so there is no need to code them in the property.

Source program file and Catalog file

There is a correspondence between DB2 table INCLUDES specified in the source program file and the tables contained into Catalog file.

When this correspondence does not exist is because a VIEW is being used. In that case, tables used are located at PlanTable file and those tables found there are those to be searched into Catalog file.

Also, although is not a good practice, it's not mandatory to INCLUDE a used table if its host variables will not be referenced. In this case, above mentioned correspondence will not exist.

DB2 Plan Table file

This file contains performance information related to DB2 accesses into the program to be analyzed.

For each access, this file contains one or more lines, and all of them belonging to the same access will have the same value in QUERYNO field.

In case the program does not access DB2, this file will be empty.

Every record is 168 characters long according to a specific format.

There are two properties defined in ${kiuwan_local_analyzer_installation_dir}/conf/analyzer.properties file to configure DB2 Plan Table file:

1. cobol.db2.plantable.path
2. cobol.db2.plantable.format

For specific information on DB2 performance aspects and EXPLAIN tables, please visit IBM related documentation pages as the following:

http://www.ibm.com/support/knowledgecenter/SSEPEK_10.0.0/perf/src/tpc/db2z_monitoranalyzedb2perfdata.html

http://www.ibm.com/support/knowledgecenter/SSEPEK_10.0.0/perf/src/tpc/db2z_createexplaintables.html

cobol.db2.plantable.path

This property defines where the DB2 PlanTable file is located. It has the same syntax as catalog file path.

An example of configuration could be:

```
cobol.db2.plantable.path=file:///@{src.dir}/resources/plantable.txt
```

cobol.db2.plantable.format

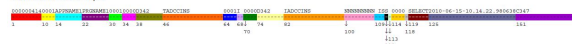This property is used to define the inner format of PlanTable file.

You have to encode the position and length (separated by ':' character) of each one of the required fields in the file. To separate each pair position:length, use the '_' character.

Take notice that there are some lengths defined between parenthesis, that means that those lengths can not be changed.
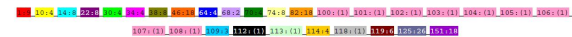
Check following examples and graphics to understand the required fields meaning and how to encode them in the property.

```
# Format definition for PlanTable resource.
cobol.db2.plantable.format=1:9_10:4_14:8_22:8_30:4_34:4_38:8_46:18_64:4_68:
2_70:4_74:8_82:18_100:(1)_101:(1)_102:(1)_103:(1)_104:(1)_105:(1)_106:(1)_107:
(1)_108:(1)_109:3_112:(1)_113:(1)_114:4_118:(1)_119:6_125:26_151:18
```

NAME	TYPE	POSITION default	LENGTH default	DESCRIPTION
QUERYNO	numeric	1	9	Query ID
QBLOCKNO	numeric	10	4	Query block num
APPLNAME	alphanumeric	14	8	Application name
PRGNAME	alphanumeric	22	8	Program name
PLANNO	numeric	30	4	Step number in query
METHOD	numeric	34	4	Method
CREATOR	alphanumeric	38	8	Owner
TNAME	alphanumeric	46	18	Table name
TABNO	numeric	64	4	Table num
ACCESSTYPE	alphanumeric	68	2	Acess type (I, I1, N, R, M, MX, MI, MU, " ")
MATCHCOLS	numeric	70	4	Number of key fields used in INDEX SCAN
ACCESSCREATOR	alphanumeric	74	8	Index creator
ACCESSNAME	alphanumeric	82	18	Index name
INDEXONLY	boolean (y/n)	100	(1)	Access by index (Y) or data (N)
SORTN_UNIQ	boolean (y/n)	101	(1)	Sorted table (SORTN) to remove duplicate rows (Y/N)
SORTN_JOIN	boolean (y/n)	102	(1)	Sorted table (SORTN) to use JOIN (Y/N)
SORTN_ORDERBY	boolean (y/n)	103	(1)	Sorted table (SORTN) by ORDER BY (Y/N)
SORTN_GROUPBY	boolean (y/n)	104	(1)	Sorted table (SORTN) by GROUP BY (Y/N)
SORTC_UNIQ	boolean (y/n)	105	(1)	Sorted table (SORTC) to remove duplicate rows (Y/N)
SORTC_JOIN	boolean (y/n)	106	(1)	Sorted table (SORTC) to use JOIN (Y/N)
SORTC_ORDERBY	boolean (y/n)	107	(1)	Sorted table (SORTC) by ORDER BY (Y/N)
SORTC_GROUPBY	boolean (y/n)	108	(1)	Sorted table (SORTC) by GROUP BY (Y/N)
TSLOCKMODE	alphanumeric	109	3	Lock mode
PREFETCH	char	112	(1)	Prefetch (L=LIST PREFETCH)
COLUMN_FN_EVAL	char	113	(1)	Column Fn Eval
MIXOPSEQ	numeric	114	4	Step order in multiplex scan operation
JOIN_TYPE	char	118	(1)	JOIN type (F, L, S, " ")
QBLOCK_TYPE	alphanumeric	119	6	Query Type
BIND_TIME	alphanumeric	125	26	BIND time
COLLID	alphanumeric	151	18	Coll Id

Source program file and Plan Table file

There is a relationship between DB2 accesses in the source program file and the information contained in PlanTable about performance for all of them.

Rows in PlanTable are grouped by AX_QUERYNO field, meaning they correspond to the same DB2 access. Rows order in PlanTable matches DB2 accesses order in the program source file. Every SELECT, UPDATE, DELETE, INSER or CURSOR declaration should be contained into PlanTable file.

Relationship between DB2 accesses (access-type objects) from source program and corresponding rows in PlanTable file is M:N according to the following table:

Considering 1:N as number of rows in PlanTable for one access.

Access Type	Access-type Objects	Relationship
Simple	1,being N the number of wos in PlanTable	1 – 1:N
With SUBSELECT	M, being M-1 the number of subselects	M - N
With UNION	M, being N the number of selects participating in the UNION	M - N
INSERT /SELECT	2, one for INSERT y one for SELECT	2 - N

Different combinations of the above table can be found depending on the access, i.e. an access with UNION and SUBSELECT will have as many access objects as SELECT in which can be decomposed. Rows identified with the same query do not match the total.