XXL tables Quality Rules enablement

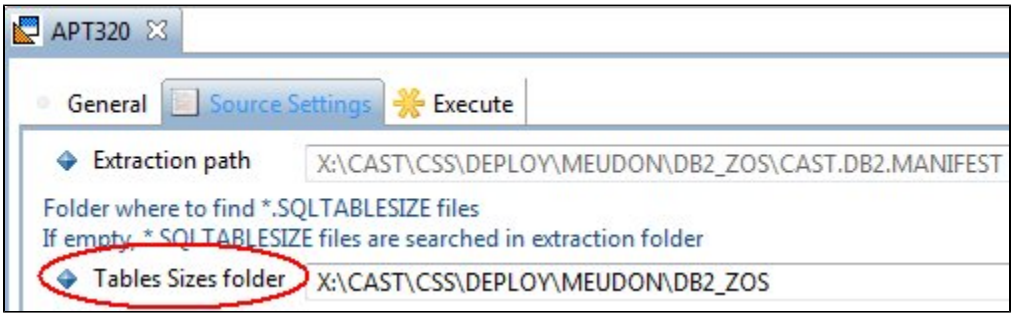
On this page:

Target audience:

CAST AI Administrators

XXL table Quality Rules are performance related Quality Rules that help detect incorrect or poorly performing SQL queries running on **XXL tables**. XXL tables can be defined as **extremely large tables**, containing a large amount of data. The threshold that determines when a table is considered to contain a large amount of data can be configured by the user using the **Assessment Models editor** in the **CAST Management Studio** (**Contextual Parameters tab**) - the default is **100,000**.

In order to enable these XXL tables Quality Rules it is necessary to provide the analyzer with table row size information. This can be done using the **Table Size Folder** option in the relevant **Analysis Unit editor** in the **CAST Management Studio**. For example, here in a **DB2 z/OS Analysis Unit**:



This folder must be populated with XML files that define your table row size information. These XML files must use the extension .SQLTABLESIZE. See examples below.

The goal is to use table size from production systems because development / integration systems may not feature really large tables and would not help detect real threat on application performance levels. If the information is not physically accessible (e.g.: first release of an application with no production environment), it is worth simulating the information by identifying tables that are expected to be large and by inputting a fake size, yet greater than the threshold in use in the "XXL tables" diagnostics. Default threshold to identify a large table is '100,000' as mentioned above.

SQLTABLESIZE file samples

These situations require the following .SQLTABLESIZE files:

Oracle Server with SQL Analyzer

```
<?xml version="1.0" encoding="UTF-8" ?>
<config name="SQL Table Size" version="1.0.0.0" extraction-date="2009/02/10" >
<oracle castformat="7.0">
<!-- "server name" is equal to the value defined for the CAST_Oracle_Instance item in the .UAXdirectory file
resulting from the extraction process -->
<server name="ORA10G" >
<schema name="CASTPUBS">
...
<table name="ORDER_LINE" rows="2000000000"/>
...
</schema>
</server>
</oracle>
</config>
```

Microsoft SQL Server with SQL Analyzer

```
<?xml version="1.0" encoding="UTF-8" ?>
<config name="SQL Table Size" version="1.0.0.0" extraction-date="2009/02/10" >
<microsoft>
<!-- server name can be written as HOST\INSTANCE_NAME but INSTANCE_NAME alone will also function -->
<server name="PDOXPLAP2\SQLS2K5" >
<!-- schema name is optional - if it is not applied, then the table row value is applied to all tables of that
name in the database -->
<schema name="schema or user name">
<database name="CASTPUBS">
...
<table name="Order_line" rows="20000000000"/>
...
</database>
</schema>
</server>
</microsoft>
</config>
```

Sybase ASE Server with SQL Analyzer

```
<?xml version="1.0" encoding="UTF-8" ?>
<config name="SQL Table Size" version="1.0.0.0" extraction-date="2009/02/10" >
<sybase>
<!-- server name can be written as HOST\INSTANCE_NAME but INSTANCE_NAME alone will also function -->
<server name="Bordeaux" >
<!-- schema name is optional - if it is not applied, then the table row value is applied to all tables of that
name in the database -->
<schema name="schema or user name">
<database name="cwmm">
...
<table name="ACC" rows="20000000000"/>
...
</database>
</schema>
</server>
</sybase>
</config>
```

IBM DB2-UDB Server

```
<?xml version="1.0" encoding="UTF-8" ?>
<config name="SQL Table Size" version="1.0.0.0" extraction-date="2009/02/10" >
<ibm-udb>
<server name="Bordeaux" >
<schema name="cwmm">
...
<table name="ACC" rows="1000000"/>
...
</schema>
</server>
</ibm-udb>
</config>
```

IBM DB2 z/OS Server

```
<?xml version="1.0" encoding="UTF-8" ?>
<config name="SQL Table Size" version="1.0.0.0" extraction-date="2009/02/10" >
<ibm-zos>
<server name="Bordeaux" >
<database name="cwmm">
<schema name="cwmm">
...
<table name="ACC" rows="1000000"/>
...
</schema>
</database>
</server>
</ibm-zos>
</config>
```

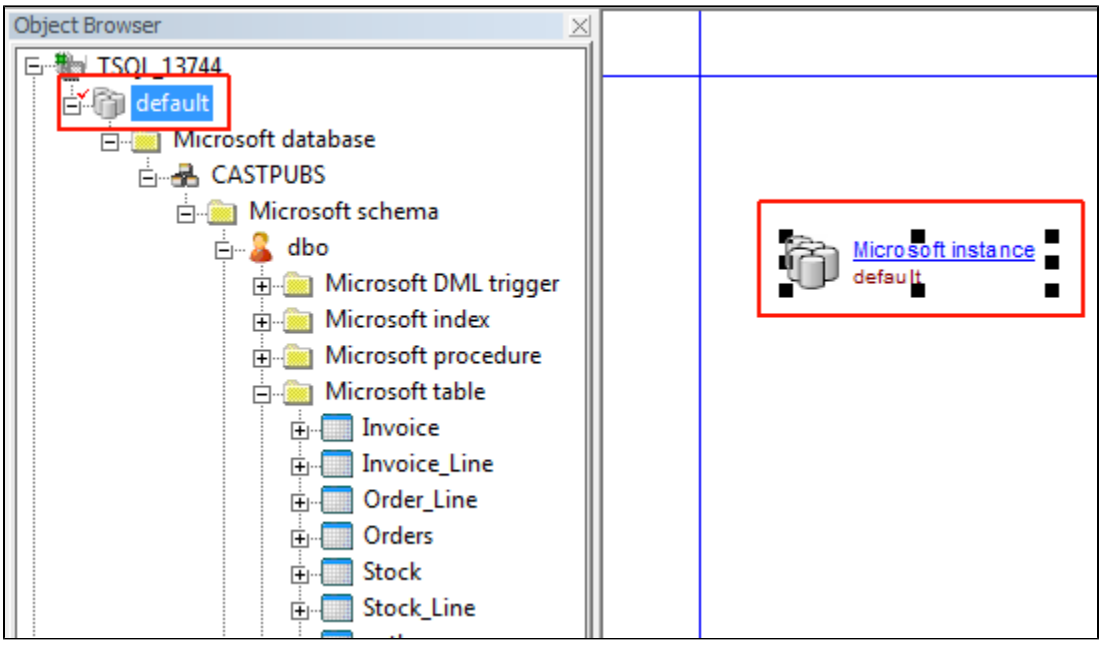
Tips - Table identification

The **server**, **database/schema** and **table** nodes in the .SQLTABLESIZE file must match the information stored in the CAST Analysis Service following an initial analysis. You can determine this information in several ways:

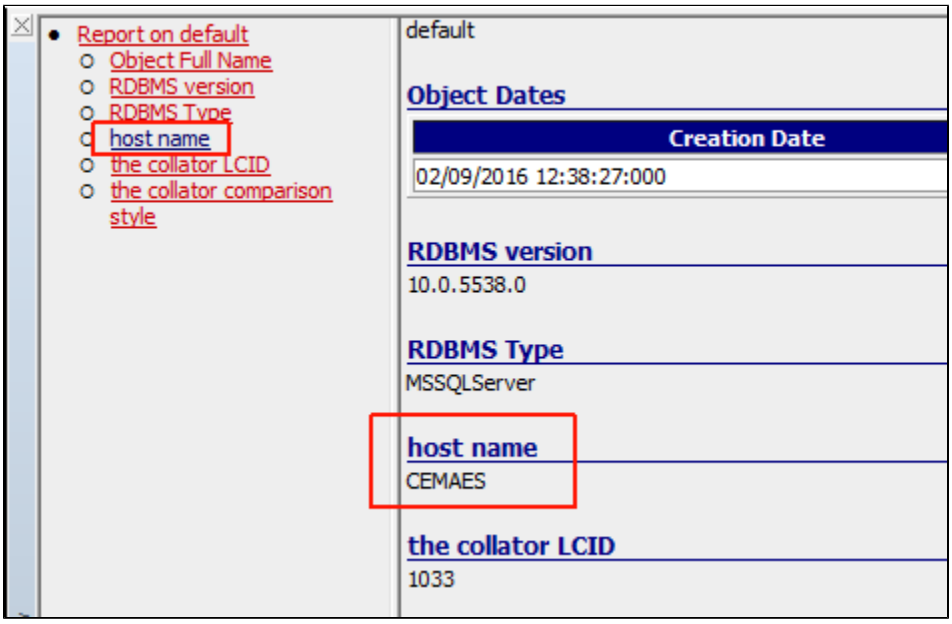
Find the server name

Using CAST Enlighten

- Right click the **Instance** either in the **Object Browser** or in the **Graphical View**:



- Select **Properties** to update the Property window. The "**host**" name highlighted below can be used to define the "**Server**" node in the . SQLTABLESIZE file:



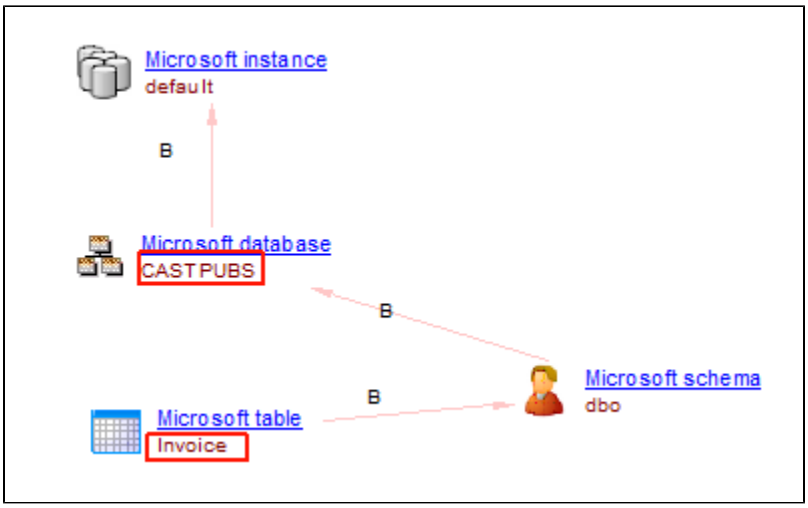
Using CAST System Views

You can query the CAST System Views in the Analysis Service to identify the "server" name. Run the following query (this is customised for CSS servers, but can be adapted to Microsoft SQL Server and Sybase ASE):

```
select DESCRIPTION
from <analysis_service>.CSV_OBJECT_DESCRIPTIONS
where DESC_TYPE = 'host name'
```

Find the Database and Table name

Databases/Schemas and **Tables** can be easily identified in CAST Enlighten using the **Object Browser** or the **Graphical View**:



SQLTABLESIZE file generation

You can generate these files by executing the following scripts:

> If you are collecting table size from a production system while analyzing a development system, please remember to change the server name value in the generated file.

Oracle - SQL Analyzer

Execute the following script using the owner of the schema you are interested in:

```
set echo off;
set heading off;
set feedback off;
spool ORACLE.SQLTABLESIZE;
select '<?xml version="1.0"encoding="UTF-8" ?>' from dual;
select '<config name="SQL Table Size" version="1.0.0.0" extraction-date="' ||to_char(sysdate, 'YYYY' )||'
/'||to_char(sysdate, 'MM' ) ||'/'||to_char(sysdate, 'dd' )||'" >' from dual;
select '<oracle castformat="7.0">' from dual;
select '<server name="'|| sys_context('USERENV', 'INSTANCE_NAME')||'" >' from dual;
select '<schema name="'||sys_context('USERENV', 'CURRENT_SCHEMA')||'">' from dual;
select '<table name="'||table_name||'" rows="'||nvl(num_rows,0) ||'"/>' from user_tables ;
select '</schema>' from dual;
select '</server>' from dual;
select '</oracle>' from dual;
select '</config>' from dual;
spool off;
exit;
```

You can also automate it as follows:

```
sqlplus <owner>/<password>@<connect identifier> @<script name>
```

Microsoft SQL Server

Execute the following script against the database you are interested in:

```
set nocount on
select '<?xml version="1.0" encoding="UTF-8" ?>'
select '<config name="SQL Table Size" version="1.0.0.0" extraction-date="' + convert(varchar(10),GETDATE()) +
'" >'
select char(9) + '<microsoft>'
select char(9) + char(9) + '<server name="'+ @@servername +'">'
select char(9) + char(9) + char(9) + '<database name="'+ DB_NAME() +'">'
select char(9) + char(9) + char(9) + char(9) +'<table name="' + so.name + '" rows="' + convert(varchar(10),
convert(int,MAX(si.rows))) +'"/>'
FROM
sysobjects so,
sysindexes si
WHERE
so.xtype = 'U'
AND
si.id = so.id
GROUP BY
so.name
select char(9) + char(9) + char(9) + '</database>'
select char(9) + char(9) + '</server>'
select char(9) + '</microsoft>'
select '</config>'
go
```

You can automate it as follows:

```
sqlcmd -U sa -P <password> -S <server> -H <host> -d <target database> -i <script> -o SQLSERVER.SQLTABLESIZE -h
-1
```

Note that if you run the above query in **Microsoft SQL Server Management Studio**, the default output mode cannot easily be copied out of the results window. Therefore CAST recommends that you change the query output to either **Text** or **File** as follows:

- Click the **Query** menu (available when working in the Query window)
- Select **Results To** and then choose either:
  - **Results To Text**
  - **Results To File**

Sybase ASE

Execute the following script against the database you are interested in:

```
select "<?xml version="1.0" encoding="UTF-8" ?>"
select "<configSQL Table Size"" version=""1.0.0.0"" extraction-date="""|| convert(varchar(12),getdate()) ||
"""">"
select char(9) ||"<sybase>"
select char(9) || char(9) || "<server name="'+ @@servername +'">"
select char(9) || char(9) || char(9) || "<database>"
begin SELECT "<table" || o.name || """ rows="""" || convert(varchar(50),convert(int,s.rowcnt)) || """ />"
FROM sysobjects o, systabstats s
WHERE o.id = s.id AND s.indid IN (0,1)
AND o.type = 'U'
ORDER BY o.type, o.name end
select char(9) || char(9) || char(9) || "</database>"
select char(9) || char(9) || "</server>"
select char(9) || "</sybase>"
select "</config>"
go
```

IBM DB2 UDB

Execute the following script against the schema you are interested in:

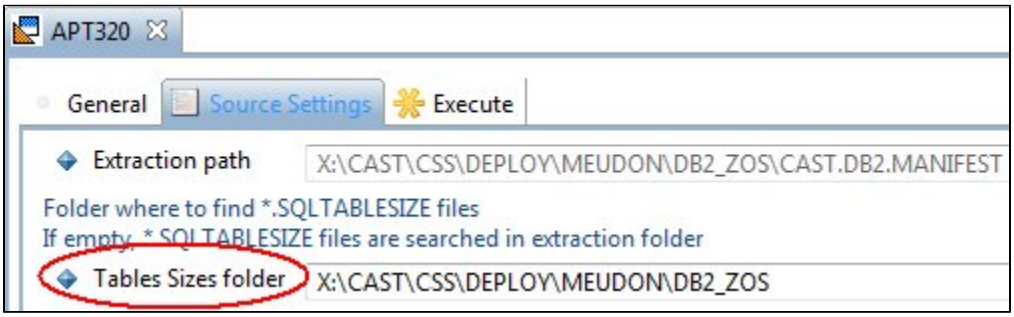
```
connect to <server> user <user> using <role>;
select '<?xml version="1.0" encoding="UTF-8" ?>' from SYSIBM.SYSDUMMY1;
select '<config name="SQL Table Size" version="1.0.0.0" extraction-date="' || varchar(current date) || '" >'
from SYSIBM.SYSDUMMY1;
select CAST(char(' ') as char(4)) || '<ibm-udb>' from SYSIBM.SYSDUMMY1;
select CAST(char(' ') as char(8)) || '<server name="' || HOST_NAME || '">' FROM TABLE(SYSPROC.
ENV_GET_SYS_INFO()) AS SYSTEMINFO;
select CAST(char(' ') as char(12)) || '<database name="'DB2UDB'">' from SYSIBM.SYSDUMMY1;
select CAST(char(' ') as char(16)) || '<schema name="CASTPUBS">' from SYSIBM.SYSDUMMY1;
select CAST(char(' ') as char(20)) || '<table name="' || tabname || '" rows="' || TRIM(TRAILING FROM CAST(card
as char(128))) || '" />' FROM syscat.tables where tabschema = 'CASTPUBS' and type = 'T' order by tabname;
select CAST(char(' ') as char(16)) || '</schema>' from SYSIBM.SYSDUMMY1;
select CAST(char(' ') as char(12)) || '</database>' from SYSIBM.SYSDUMMY1;
select CAST(char(' ') as char(8)) || '</server>' from SYSIBM.SYSDUMMY1;
select CAST(char(' ') as char(4)) || '</ibm-udb>' from SYSIBM.SYSDUMMY1;
select '</config>' from SYSIBM.SYSDUMMY1;
```

Please ensure that you:

- modify the first line to use your own login credentials.
- you will need to run this script against each schema you are interested in and for each schema you will need to modify the lines containing:
  - <schema name="CASTPUBS">
  - where tabschema = 'CASTPUBS'
- The resulting DB2UDB.SQLTABLESIZE file is not valid XML, please ensure that you remove the first few lines that are not part of the valid generated XML.

IBM DB2 z/OS

The DB2 z/OS extractor includes script (**STEP 26**) to generate a "raw" file that will contain the number of rows in each table - i.e. sufficient information for defining an .SQLTABLESIZE file. This file is referenced in the DB2 z/oS MANIFEST generated by the extractor. When configuring the analysis in the CAST Management Studio, you simply need to input the location of the **folder** that contains the MANIFEST file into the **Table Sizes folder** field:



In the example above, the MANIFEST has been deployed to a location in the Deployment folder (**Extraction path** field). The **Table Sizes folder** field has then been set to the same folder that contains the MANIFEST file. When the analysis is run, the analyzer will automatically generate an .SQLTABLESIZE file based on the raw table size file generated during the extraction and which is referenced in the MANIFEST. This raw file will be used to generate the required data for display in the CAST AIP dashboards. Therefore, you do not need to manually generate the SQLTABLESIZE file yourself - the analyzer will do this for you provided that you enter the required folder into the **Table Sizes folder** field.

Note that:

- you can still manually define your own SQLTABLESIZE file based on the example given above, however, CAST recommends using the "automatic" generation method in the CAST Management Studio.
- by default the DB2 z/OS extractor is set to output the raw file containing table size information with the name "CAST.DB2.TABROWS". However, the extractor JCL may have been manually modified and the output file may be named differently or, during the transfer to the Windows environment, the resulting file may be named differently. In all cases, you should ensure that the MANIFEST file lists the file that contains the raw table size information and that the name referenced in the MANIFEST is identical to the file that has been delivered via the CAST Delivery Manager Tool

Troubleshooting

Checking SQLTABLESIZE information upload

To know which table has been tagged with the SQLTABLESIZE information, you can run on local site the following query:

```
Select OBJECT_NAME, OBJECT_FULLNAME, OI.InfVal
From CDT_OBJECTS CO, ObjInf OI
Where OI.IdObj = CO.OBJECT_ID
And OI.InfTyp = 115
And OI.InfSubTyp = 1
```

Finding XXL tables

To know which are the XXL tables, , you can run on local site the following query:

```
Select OBJECT_NAME, OBJECT_FULLNAME, OI.InfVal
From CDT_OBJECTS CO, ObjInf OI
Where OI.IdObj = CO.OBJECT_ID
And OI.InfTyp = 115
And OI.InfSubTyp = 1
And OI.InfVal >= 100000 -- change the value if you updated the threshold in the Assessment Model
```