Version 22

Custom SQL Reporting

Contents

Introduction

This guide is intended for Administrators and Operators who configure or maintain Custom SQL reports. Additional Bocada product documentation is publicly available on the Bocada Support web site https://bocada-support.force.com/.

Functionality Overview

Bocada has basic and advanced capabilities for Custom SQL Reporting available in two modules: *SQL Queries* (basic) and *Custom SQL Templates* (advanced). The following is available for each:

Custom SQL Templates Capabilities Only

Custom SQL Templates have distinct advanced capabilities:

- Criteria Filtering. The user viewing the report will be able to use standard Bocada UI filters, just as they can for all Bocada tabular reports.
- Ability to SUM columns
- Scheduled Report Distribution

Custom SQL Templates and SQL Queries Capabilities

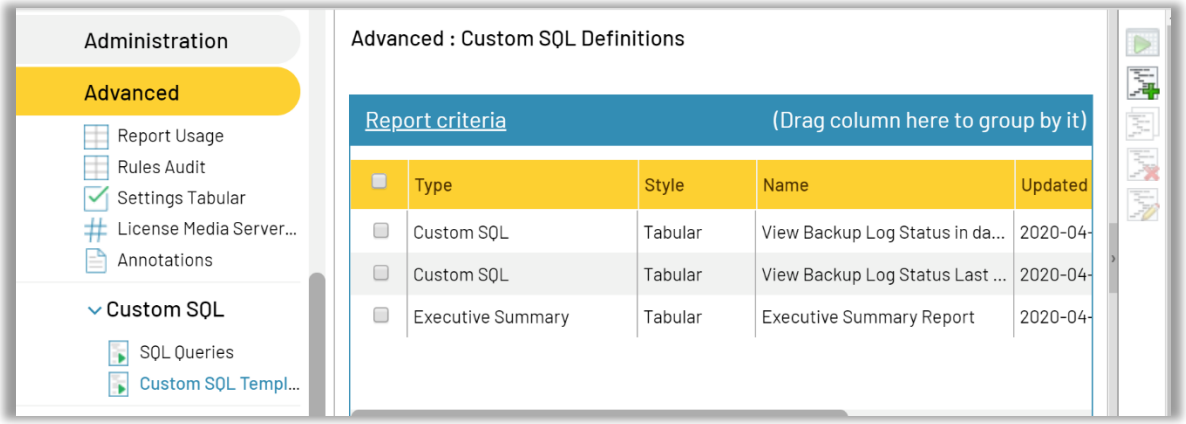
- SELECT queries against tables and views *
- EXECUTE stored procedures
- Using #TEMP tables
- Column grouping and sorting
- Save Report, Open Saved Report
- Send as Email
- Download as File

Custom SQL Templates

Custom SQL Report templates provide a more feature rich experience including criteria and the ability to save and schedule. Administrators can create, run, and manage all Custom SQL report templates and their resultant saved reports and distribution.

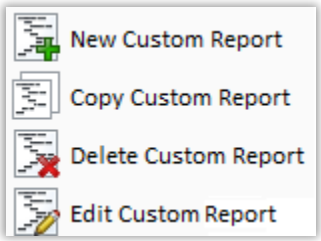
Once a custom report template is added via this view, then users can see, run, and save Custom SQL reports with their own criteria via the Saved Reports view. This is the only place that saved versions of a custom report type can be run.

You can add and manage custom SQL report templates, as well as run the base template, in the Custom SQL Templates view.

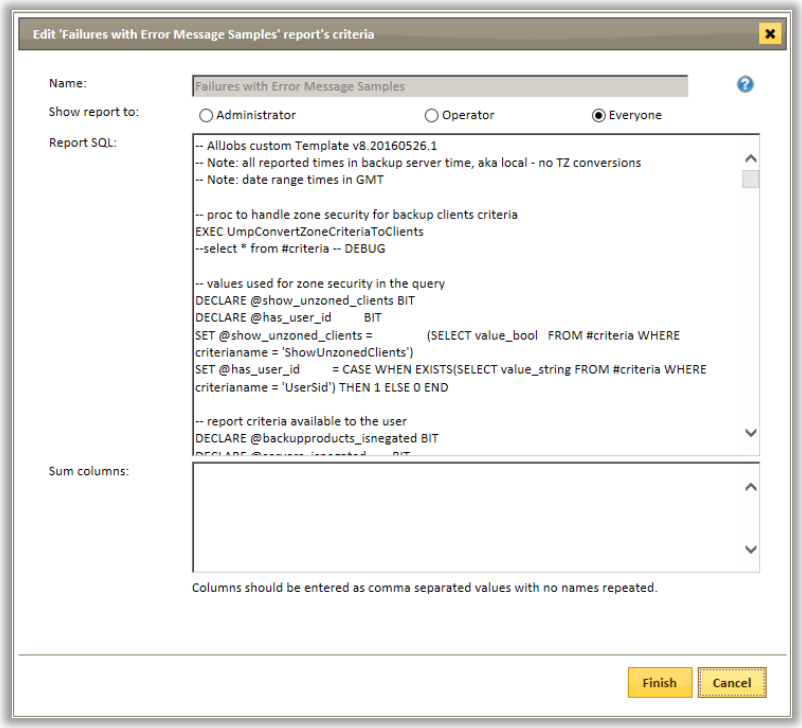


Add a New Custom SQL Report Template

Writing custom SQL for Bocada is covered in *Appendix C: Writing Your Own Custom SQL Reports*. Once you have the SQL code for a new report, perhaps from Bocada Professional Services, then you can add it to your Bocada Deployment by using these actions:



Click New or Edit Custom Report icon button to see the dialog:



Give the template a name, set the custom report accessibility with the radio buttons. Paste your SQL text into the Report SQL text box. When columns are added to the 'Sum columns' section, the report will appear as before but with those columns first and the sum of those columns' values at bottom.

Click Finish, and then run your base template report!

Only administrators can edit the SQL code. When access is granted to non-administrator roles, they will only be able to run the report from the Saved Reports view and edit and save and with their own criteria.

Saved Custom SQL Templates will appear under Saved Reports as Report Type *Custom SQL* as Templates.

Criteria Filters

It is possible to write criteria filters into Custom SQL Templates which can be set by users through the same mechanisms of standard Bocada reports. These report templates can then be managed through the Saved Reports interface, run, saved with different criteria filtering applied, and distributed on schedules. For example, Custom SQL reports may be created from the same template with different criteria for different end users.

Standard Criteria Filters

Criteria Filter	Keyword	Criteria Type	Example
Date Range	$FromDate	DATETIME	`SELECT * FROM v_backuplogall WHERE jobdatetimelocal > $FromDate`
	$ToDate	DATETIME	`SELECT * FROM v_backuplogall WHERE jobdatetimelocal < $ToDate`
Backup Products	$BackupProducts	List of IDs	`SELECT * FROM v_backuplogall WHERE product_id IN ($BackupProducts)`
Backup Servers	$Servers	List of IDs	`SELECT * FROM v_backuplogall WHERE server_id IN ($Servers)`
Backup Clients	$Clients	List of IDs	`SELECT * FROM v_backuplogall WHERE client_id IN ($Clients)`
Job Groups	$JobGroups	List of IDs	`SELECT * FROM v_backuplogall WHERE backupgroup_id IN ($JobGroups)`
Backup Levels	$Levels	List of IDs	`SELECT * FROM v_backuplogall WHERE level_id IN ($JobGroups)`
Zones	$Zones	List of IDs	`SELECT * FROM v_backuplogall bl JOIN backupclientzone bcz WITH (NOLOCK) ON bcz.client_id=bl.client_id JOIN zone z WITH (NOLOCK) ON z.zone_id=bcz.zone_id WHERE z.zone_id IN ($Zones)`
Media Libraries	$MediaLibraries	List of IDs	`SELECT * FROM medialog mg JOIN mediadevice md WITH (NOLOCK) ON md.mediadevice_id=mg.mediadevice_id JOIN medialibrary ml WITH (NOLOCK) ON ml.medialibrary_id=md.medialibrary_id WHERE ml.medialibrarypreferred_id IN ($MediaLibraries)`
Media Servers	$MediaServers	List of IDs	`SELECT * FROM medialog WHERE mediaserver_id IN ($MediaServers)`

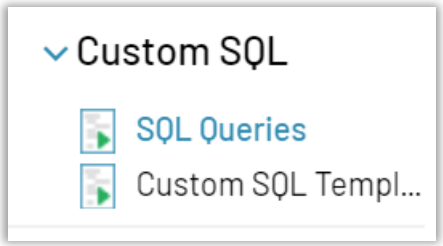
Additional Criteria Filters

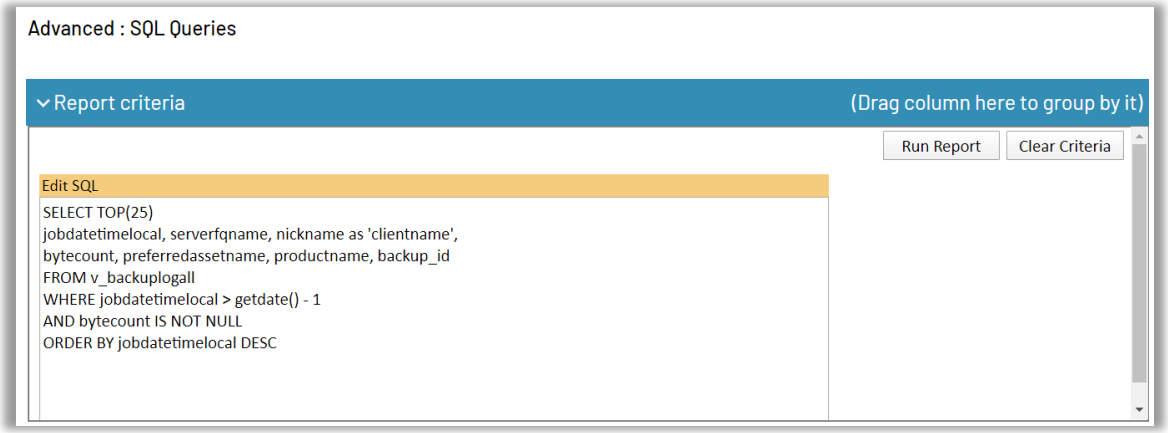
Additional criteria are available for report creation, though it is complex and more difficult to configure and has been deprecated. Additional information can be found in the Appendix. If you require filtering by criteria not available in the standard set, please provide your feedback to Bocada Support for potential addition to the standard supported list.

SQL Queries

The SQL Queries view into the Bocada database allows simple SQL statements to be quickly run against tables and views in the database.

With a SQL Query report, the criteria are limited only by what you provide in SQL:





Saved SQL Query reports will appear under Saved Reports as *Type: Admin SQL Queries*.
As with standard reports, the accessibility of saved SQL Query reports is controlled by Administrators.

Troubleshooting and Limitations

You must name unnamed columns. For example, you cannot do:

```
      SELECT 'myValue'
```

```
You must add a column name:
```

```
      SELECT 'myValue' AS 'myColumn'
```

Appendix A: Additional Legacy Method with #Criteria

Bocada supports creating custom SQL reports (as above in the SQL Queries section), and with criteria as seen below, and supports access controls so reports can be run by and shared with all users, operators, or administrators.

Running a Custom SQL Report using legacy #Criteria Method

Custom SQL reports are available to designated users through the *Saved Reports* view.

Custom Report Administration using legacy #Criteria Method

Administrators can manage custom SQL reports by using the tools found under Administration > Custom SQL Templates, above.

Date/Time Controls using legacy #Criteria Method

The following comments will toggle date / time controls when added to Custom SQL:

```
-- TimeEnableBocada -- enables time controls
-- TimeZoneEnableBocada -- enables time zone control
-- IncludeBackupServerTimeBocada -- includes backup server time as a
timezone
SET @start_date = CONVERT(VARCHAR, (SELECT value_datetime FROM
#criteria WHERE criterianame = 'FromDatetime'), 101)
SET @end_date = CONVERT(VARCHAR, (SELECT value_datetime FROM #criteria
WHERE criterianame = 'ToDatetime'), 101)
```

Report Criteria using legacy #Criteria Method

The table below lists the optional criteria that can be used in a report. Bocada will automatically detect which criteria are provided to the user if the report follows the rules prescribed below.

Criterion	**Value Type** ++ = see notes below	**Multi-value**	**Notes**
`UserSid`	`value_string ++`		This is not a control but an automatically supplied value equal to the SID of the user.

`FromDateTime`
`ToDateTime`
`TimeZone`
`= Date Range`
	`value_datetime`
	`value_datetime`
	`value_bigint`
	`[ztimezone.timezone_id]`
			The value of *ToDateTime* will be the date of one day past the date selected by the user if only a date is specified, making the date range inclusive of the start and end days.

`BackupProducts`	`value_bigint` `[backupproducts.product_id]`	Yes	
`BackupLevels`	`value_string`	Yes	
`BackupServers`	`value_bigint` `[servers.server_id]`	Yes	
`BackupClients`	`value_bigint` `[clients.client_id]`	Yes	
`Assets`	`value_bigint [asset.asset_id]`	Yes	
`JobGroups`	`value_bigint` `[backupgroups.backupgroup_id]`	Yes	
`MediaServers`	`value_bigint` `[mediaserver.mediaserver_id]`	Yes	
`MediaLibraries`	`value_bigint` `[medialibrary.medialibrary_id]`	Yes	
`Zones`	`value_bigint [zone.zone_id]`	Yes	
`SLAProfiles`	`value_bigint` `[slaprofile.slaprofile_id]`	Yes	
`PriceLists`	`value_bigint` `[pricelist.pricelist_id]`	Yes	
`Attempt` = Backup Attempts	`value_bigint ++`		1 = best, 2 = last, 3 = all
`IncludeIgnoredItems`	`value_bool`		

Processing Simple Criteria using legacy #Criteria Method

Below is an example showing how to extract single values from specified criteria. To extract single values from the criteria use this code:

```
SET @criterion_value =
    SELECT value_type FROM #criteria
    WHERE criterianame = 'criterion'
```

The ***value_type*** and ***criterion*** are user-specified; The example above will result in No Data when the report is run unless these are changed to real values.

Processing Multi-Valued Criteria

All multi-valued criteria support selecting by either inclusion or exclusion. Therefore, to use these criteria, the report needs to first check if the set is negated using this code:

```
SET @criterion_isnegated =
    SELECT TOP 1 isnegated FROM #criteria
    WHERE criterianame = 'criterion'
```

Based on the result, the criteria values either identify the values included or excluded. To specify exclude/include in a report query, use the following example edited for your purposes:

```
AND (@criterion_isnegated IS NULL
     OR @criterion_isnegated = 0
        AND id IN (SELECT value_bigint FROM #criteria
                   WHERE criterianame = 'criterion')
     OR @criterion_isnegated = 1
        AND id NOT IN (SELECT value_bigint FROM #criteria
                       WHERE criterianame = 'criterion')
```

Zones and Backup Clients using legacy #Criteria Method

WARNING: Your SQL code is responsible for implementing zone security. To ensure that users will not have access to clients that they are not authorized to see, you must follow the instructions in this section. There is nothing in place to prevent a report from displaying data from zones that the user does not have access to; this is entirely the responsibility of the report author.

To enforce zones, the report must include this at the top of the SQL code to set things up:

```
EXEC dbo.UmpConvertZoneCriteriaToClients
DECLARE @show_unzoned_clients INT
DECLARE @has_user_id INT
SET @show_unzoned_clients =
    (SELECT value_bool FROM #criteria
     WHERE criterianame = 'ShowUnzonedClients')
SET @has_user_id =
    CASE WHEN EXISTS(SELECT value_string FROM #criteria
                     WHERE criterianame = 'UserSid')
    THEN 1 ELSE 0 END
```

Next, add this condition to your SQL query:

```
(@has_user_id = 0
 OR clients.client_id IN
     (SELECT DISTINCT clients.client_id
      FROM clients WITH (NOLOCK)
      LEFT JOIN backupclientzone AS bcz WITH (NOLOCK)
          ON bcz.client_id = clients.client_id
      LEFT JOIN zwebuserzone AS wuz WITH (NOLOCK) ON wuz.zone_id =
bcz.zone_id
      LEFT JOIN zwebusers AS wu WITH (NOLOCK) ON wu.user_id = wuz.user_id
      LEFT JOIN #criteria ON #criteria.value_string = wu.sid
                          AND #criteria.criterianame = 'UserSid'
      WHERE (#criteria.value_string IS NOT NULL OR bcz.client_id IS NULL)
             AND (@show_unzoned_clients = 1 OR bcz.zone_id IS NOT  NULL)))
```

See the sample report below to see how zones are used in a report. Note that this code does not enable the Zones criteria to the user. To enable zones without actually querying directly for zones, add this comment to your SQL query:

```
-- criterianame = 'Zones'
```

Error Handling using #Criteria Method

If the Custom SQL report returns “No Data” when run, you can verify your custom SQL query using the SQL query analyzer built into Microsoft SQL Server or contact your SQL administrator for further troubleshooting steps.

Custom Report for Debugging using #Criteria Method

Here is a custom report that is particularly useful for testing different criteria to demonstrate the #criteria table looks like when your report is run:

```
SELECT * FROM #criteria
WHERE criterianame = 'BackupClients'
    OR criterianame = 'BackupServers'
```

Add as many lines as necessary, listing each of the criteria names you want to use in your report. If you don’t set any criteria values, you will see “No Data”, as the criteria table does not include any values for unset criteria.

Example Custom Report using #Criteria Method

The following example Custom SQL report will show information about backup servers and the time of the last backup, as well as how to implement zone security.

```
-- Call proc to handle zone security for backup clients criteria.
EXEC UmpConvertZoneCriteriaToClients

-- These values are used for zone security in the query.
DECLARE @show_unzoned_clients INT
DECLARE @has_user_id INT
SET @show_unzoned_clients =
    (SELECT value_bool FROM #criteria
     WHERE criterianame = 'ShowUnzonedClients')
SET @has_user_id =
    CASE WHEN EXISTS(SELECT value_string FROM #criteria
                     WHERE criterianame = 'UserSid')
    THEN 1 ELSE 0 END

-- These values are the report criteria available to the user.
DECLARE @backupproducts_isnegated INT
DECLARE @servers_isnegated INT
DECLARE @clients_isnegated INT
DECLARE @from_date_time DATETIME
DECLARE @to_date_time DATETIME

SET @backupproducts_isnegated =
    (SELECT TOP 1 isnegated FROM #criteria
     WHERE criterianame = 'BackupProducts')
SET @servers_isnegated =
    (SELECT TOP 1 isnegated FROM #criteria
     WHERE criterianame = 'BackupServers')
SET @clients_isnegated =
    (SELECT TOP 1 isnegated FROM #criteria
     WHERE criterianame = 'BackupClients')
SET @from_date_time =
    (SELECT value_datetime FROM #criteria
     WHERE criterianame = 'FromDatetime')
SET @to_date_time =
    (SELECT value_datetime FROM #criteria
     WHERE criterianame = 'ToDatetime')

SELECT
    clients.clientfqname AS 'Client',
    clients.lastjobdatetime AS 'Last Backup Date Time',
    backupproducts.productname AS 'Backup Product',
    servers.serverfqname AS 'Server'
FROM clients
JOIN servers ON servers.server_id = clients.server_id
            AND servers.serverstatus <> 'delete'
JOIN backupproducts ON backupproducts.product_id = servers.product_id
WHERE clients.client_id > 0

    -- Filter by user-selected date range; note that lastjobdatetime is in
UTC.
    AND clients.lastjobdatetime IS NOT NULL
```

```
    AND clients.lastjobdatetime >= @from_date_time
    AND clients.lastjobdatetime < @to_date_time

    -- Filter by user-selected backup product ids.
    AND (@backupproducts_isnegated IS NULL
         OR @backupproducts_isnegated = 0
            AND servers.product_id IN
                (SELECT value_bigint FROM #criteria
                 WHERE criterianame = 'BackupProducts')
         OR @backupproducts_isnegated = 1
            AND servers.product_id NOT IN
                (SELECT value_bigint FROM #criteria
                 WHERE criterianame = 'BackupProducts'))

    -- Filter by user-selected server ids.
    AND (@servers_isnegated IS NULL
         OR @servers_isnegated = 0
            AND clients.server_id IN
                (SELECT value_bigint FROM #criteria
                 WHERE criterianame = 'BackupServers')
         OR @servers_isnegated = 1
            AND clients.server_id NOT IN
                (SELECT value_bigint FROM #criteria
                 WHERE criterianame = 'BackupServers'))

    -- Filter by user-selected client ids.
    AND (@clients_isnegated IS NULL
         OR @clients_isnegated = 0
            AND clients.client_id IN
                (SELECT value_bigint FROM #criteria
                 WHERE criterianame = 'BackupClients')
         OR @clients_isnegated = 1
            AND clients.client_id NOT IN
                (SELECT value_bigint FROM #criteria
                 WHERE criterianame = 'BackupClients'))

    -- Implement zone security; this block is required for
    -- all reports accessible to non-admins.
    AND (@has_user_id = 0
         OR clients.client_id IN
             (SELECT DISTINCT clients.client_id
              FROM clients WITH (NOLOCK)
              LEFT JOIN backupclientzone AS bcz WITH (NOLOCK)
                  ON bcz.client_id = clients.client_id
              LEFT JOIN zwebuserzone AS wuz WITH (NOLOCK)
                  ON wuz.zone_id = bcz.zone_id
              LEFT JOIN zwebusers AS wu WITH (NOLOCK)
                  ON wu.user_id = wuz.user_id
              LEFT JOIN #criteria ON #criteria.value_string = wu.sid
                                  AND #criteria.criterianame = 'UserSid'
              WHERE (#criteria.value_string IS NOT NULL
                     OR bcz.client_id IS NULL)
                  AND (@show_unzoned_clients = 1
                       OR bcz.zone_id IS NOT NULL)))

ORDER BY clients.lastjobdatetime DESC
```

Technical Support

For technical support or a copy of our standard support agreement, please contact us.

E-mail: support@bocada.com
Support Portal: http://www.bocada.com/support/
Phone: +1-425-898-2400