



Upgrade From 4.2.1

Breaking changes

Grid Conditional Formatting

Before version 4.2.2, grid conditional formatting was tightly coupled with the grid functionality, forcing its implementation even when it was not intended to be used. However, with a strategic enhancement, conditional formatting has now been decoupled from the grid. As a result, the conditional formatting feature can be implemented optionally.

The `GridViewModel` object now includes the `ConditionalFormatting` property, which contains the necessary configuration to process the conditional formatting for GRID-type reports.

To fulfill the new conditional formatting call, you must **add** the following section to the existing implementation:

```
<div id="conditional-formatting-wrapper">
    @Html.InfoArch().RenderConditionalFormatting(Model.ConditionalFormatting)
</div>
```

Complete Code

```
<button onclick="IA_dRevealConditionalEditor.open(580, 520); return false;">
    Conditional Formatting
</button>

@model GridViewModel
<div id="conditional-formatting-wrapper">
    @Html.InfoArch().RenderConditionalFormatting(Model.ConditionalFormatting)
</div>
```

Note: Do not modify the existing implementation. This release only requires adding the specified portion of code.

New changes

New libraries

Two libraries have been added to support **MySQL** data source.

- `InfoArch.Common.Providers.v4.2.Document.MySQL.dll`
- `InfoArch.Common.Providers.v4.2.MySQL.dll`

Note: These DLLs are optional. If you do not use the Mysql data source, you do not need to add these DLLs.

Web.config file

`<dReveal />` tag has added two sections to manage **SQL Server** data source provider and **MySQL** data source provider.

Data Sources

```
<sqlserver
  executionFormat="Inherit|Nested|Cte"
  paginationType="Index|OffsetFetch"
  cleanColumns="false|true"
/>
```

! INFO

[Subqueries](#) and [common table expression](#) in SQL Server.

```
<mysql
  executionFormat="Inherit|Nested|Cte"
  cleanColumns="false|true"
/>
```

! INFO

[Subqueries](#) and [common table expression](#) in MySQL.

Attribute	Description
executionFormat	<p>The type of formatting applied to the SQL query for the report before it is executed.</p> <ol style="list-style-type: none"> 1. Nested: This format uses subqueries to construct SQL queries for reports. 2. CTE: This format uses common table expressions to construct SQL queries. 3. Inherit: This inherits the format type (Nested, CTE) that was configured in the model designer. <p>Default: Inherit.</p>
cleanColumns	<p>Sets the setting to remove unused columns in the database query.</p> <ol style="list-style-type: none"> 1. true: Remove unused columns from the database query. 2. false: Do not remove unused columns from the database query. <p>Default: false.</p>
paginationType	<p>The strategy is used for paginating Grid reports. Only support SQL Server.</p> <ol style="list-style-type: none"> 1. Index: It utilizes the SQL Server's ROW_NUMBER function for pagination. 2. OffsetFetch: It utilizes the SQL Server's OFFSET/FETCH clause for pagination. <p>Default: Index.</p>

DRevealGlobalSettings class

EnabledPivotExportExcelObject

This is a new property that enables or disables the creation of a pivot object within an Excel file when set to true, otherwise it only creates simple information showing the total values.

Default value: true

Definition

```
public bool EnabledPivotExportExcelObject { get; set; }
```

Example

```
public ReportingController()
{
    DRevealGlobalSettings.Instance.EnabledPivotExportExcelObject = true;
}
```

Note: The use of this property is optional.

GridViewModel class

SetCleanColumnsOption

This method enables or disables the cleaning of unnecessary columns in the SQL query.

Definition

```
public void SetCleanColumnsOption(bool clean)
```

Note: The use of this method is optional.

DashboardViewModel class

SetCleanColumnsOption

This method enables or disables the removal of unnecessary columns in the SQL query.

Definition

```
public void SetCleanColumnsOption(bool clean)
```

EnablePivotExportExcelObject

This method enables or disables the creation of a pivot object within an Excel file when set to true, otherwise it creates simple data displaying only the total values.

Definition

```
public void EnablePivotExportExcelObject(bool enable)
```

SetMaxCharsLength

This method sets the maximum character length allowed for bar chart axis labels.

Definition

```
public void SetMaxCharsLength(int maxCharsLength)
```

Note: The use of these methods is optional.

Excel export

To exporting an Excel file, please note the following notes when naming Excel sheets:

- The Excel sheet name must not be equal to an existing name in the collection of worksheets.
- The Excel sheet name must not exceed 31 characters.
- The Excel sheet name must not contain the following symbols: /, ?, :, *, [or]
- The Excel sheet name must not start and end with a single quote (').
- The Excel sheet name must not be an empty string.