SSRS: Export Reports Without WebPortal

If you’re trying to export all the RDL, data sources, gif, and other files of a set of reports, but the SSRS web portal is inaccessible, this method will pull them from the report engine database back-end.

Taken from here: Export of SSRS reports datasources and images

```
<# .SYNOPSIS
        Export of SSRS reports datasources and images
    .DESCRIPTION
        This PowerShell script exports all (or filtered) reports, data sources and images directly from the ReportServer database
        to a specified folder. For the file name the complete report path is used; for file name invalid characters are replaced with a -.
        Reports are exported with .rdl as extension, data sources with .rds and resources without any additional extension.
        Please change the "Configuration data" below to your enviroment.
        Works with SQL Server 2005 and higher versions in all editions.
        Requires SELECT permission on the ReportServer database.
    .NOTES
        Author  : Olaf Helper
        Requires: PowerShell Version 1.0, Ado.Net assembly
    .LINK
        GetSqlBinary: http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqldatareader.getsqlbinary.aspx
#>
param(
        [Parameter(Mandatory=$true)] [string]$serverName, # server name and instance
        [Parameter(Mandatory=$true)] [string]$databaseName, # ReportServer Database.
    [Parameter()] [string]$outputDirectory = 'C:\Temp\' # Path to export the reports to.
)

# Select-Statement for file name & blob data with filter.
```

```
$sql = "SELECT CT.[Path]
              ,CT.[Type]
              ,CONVERT(varbinary(max), CT.[Content]) AS BinaryContent
       FROM dbo.[Catalog] AS CT
       WHERE CT.[Type] IN (2, 3, 5)";

# Open ADO.NET Connection with Windows authentification.
$con = New-Object Data.SqlClient.SqlConnection;
$con.ConnectionString = "Data Source=$serverName;Initial Catalog=$databaseName;Integrated
Security=True;";
$con.Open();

Write-Output ((Get-Date -format yyyy-MM-dd-HH:mm:ss) + ": Started ...");

# New command and reader.
$cmd = New-Object Data.SqlClient.SqlCommand $sql, $con;
$rd = $cmd.ExecuteReader();

$invalids = [System.IO.Path]::GetInvalidFileNameChars();
# Looping through all selected datasets.
While ($rd.Read())
{
   Try
   {
      # Get the name and make it valid.
      $name = $rd.GetString(0);
      foreach ($invalid in $invalids)
         {    $name = $name.Replace($invalid, "-");    }

      If ($rd.GetInt32(1) -eq 2)
         {    $name = $name + ".rdl";    }
      ElseIf ($rd.GetInt32(1) -eq 5)
         {    $name = $name + ".rds";    }

      Write-Output ((Get-Date -format yyyy-MM-dd-HH:mm:ss) + ": Exporting {0}" -f $name);

      $name = [System.IO.Path]::Combine($outputDirectory, $name);

      # New BinaryWriter; existing file will be overwritten.
      $fs = New-Object System.IO.FileStream ($name), Create, Write;
```

```
        $bw = New-Object System.IO.BinaryWriter($fs);

        # Read of complete Blob with GetSqlBinary
        $bt = $rd.GetSqlBinary(2).Value;
        $bw.Write($bt, 0, $bt.Length);
        $bw.Flush();
        $bw.Close();
        $fs.Close();
    }
    Catch
    {
        Write-Output ($_.Exception.Message)
    }
    Finally
    {
        $fs.Dispose();
    }
}

# Closing & Disposing all objects
$rd.Close();
$cmd.Dispose();
$con.Close();
$con.Dispose();

Write-Output ((Get-Date -format yyyy-MM-dd-HH:mm:ss) + ": Finished");
```

Revision #1
Created 10 June 2025 05:25:45 by glwhite
Updated 1 September 2025 05:18:38 by glwhite