

SQL Stored Procedure for Custom CSV Filename

Introduction

When using the RDS Intranet CSV data export tool, it is sometimes desirable to create a file name different than what the export routine generates. The CSV export tool supports this capability by calling a user specified SQL Stored Procedure. This note defines the passed parameters the procedure must have and provides an example of such a stored procedure.

When a stored procedure is used, the following options are ignored by the export routine. However, the stored procedure can (and most likely should) still use them by querying the selected CSV_Names record via the passed in CSV_ID database key.

- Separate Folders
- Extension
- Add extra lot info
- Path

SQL Stored Procedure Details

Besides the filename (with an extension and no trailing “\”), the stored procedure must also return the export path in the ExportDir parameter.

The CSV export tool requires the stored procedure to have the following passed parameters in the following order:

- @Lot_ID Int
- @Wafer_ID Int
- @CSV_ID Int
- @ExportDir nvarchar(260)* Output
- @FileName nvarchar(500) Output

**260 length is for RDS Intranet 1.31, patch level RSP10052 or higher. Older versions length is 50.*

SQL Stored Procedure Example

Table 1 is an example SQL stored procedure that obtains the path and extension from the options set on CSV export tool configuration screen. It also appends a date-time stamp to each filename.

```

CREATE PROCEDURE CustomCSVFileName(@Lot_ID Int, @Wafer_ID Int, @CSV_ID Int,
                                   @ExportDir nvarchar(260) Output,
                                   @FileName nvarchar(500) Output)
AS
/*
|-----
| PROCEDURE : CustomCSVFileName
| VERSION   : 1.02
| DATE      : February 12, 2010
|-----
| DEFINITION/NOTES:
|   Based upon the passed in DB keys, generates the
|   filename for the CSV export in this format:
|   LotName_IntradiePatternName_WaferName_WaferOrder_Date
|   Lot_ID    - Unique key for the DB lot
|   Wafer_ID  - Unique key for the wafer ID. If Wafer_ID
|               is 0, then wafer name is blank.
|   CSV_ID    - Unique key for the export options,
|               used to get the file extension
|   ExportDir - is returned path to export to.
|   FileName  - is the returned file name.
|-----
*/

Declare @TstN as nvarchar(30)      /* Var for max DB intradiepattern name */
Declare @LotN as nvarchar(60)     /* Var to handle max DB lot name size */
Declare @WafN as nvarchar(60)    /* Var to handle max DB wafer name size */
Declare @WctN as varchar(5)      /* Var for wafer order # */
Declare @Fdel As char(1)         /* Delimiter between filename items */
Set @Fdel = '_'

/* Init name vars in case of read error */
Set @TstN='' Set @LotN='' Set @WafN='x' Set @WctN='0'

/* Get Lot_Name used in the lot */
Select @LotN=Lot_Name From Tested_Lots Where Lot_ID=@Lot_ID

/* Get the name of the IntradiePattern used in the lot */
/* Assumes first intradie pattern in die pattern is for filename */
Select @TstN=IntradiePattern_Name
From Tested_Lots, IntraDiePattern_Names, Device_Library, DiePattern_Details
Where Tested_Lots.Lot_ID=@Lot_ID And
      Tested_Lots.Device_ID=Device_Library.Device_ID And
      Tested_Lots.Device_Version=Device_Library.Device_Version And
      Device_Library.DiePattern_ID = DiePattern_Details.DiePattern_ID And
      Device_Library.DiePattern_Version = DiePattern_Details.DiePattern_Version And
      DiePattern_Details.IDP_ID1=IntradiePattern_Names.IntraDiePattern_ID

/* If Wafer_ID passed in, get it and wafer order too */
If @Wafer_ID>0
Begin
  Select @WafN=Wafer_Name From Wafer_Names Where Wafer_ID=@Wafer_ID
  Select @WctN=WaferOrder From LotWafers Where (Lot_ID=@Lot_ID) And (Wafer_ID=@Wafer_ID)
End

/* Get file extension and path from CSV configuration */
Select @ExportDir=Path, @FileName=Extension From CSV_Names Where CSV_ID=@CSV_ID

/* Generate filename, Windows file name cannot have : so that is replaced with ` */
Set @FileName=@LotN+@Fdel+@TstN+@Fdel+@WafN+@Fdel+@WctN+@Fdel+
  replace(convert(varchar,getdate(),126),':','`')+'.'+@FileName

Return(0)
GO

```

Table 1 – Example SQL Stored Procedure