



iParameter SSRS

Report Parameter Saving Solution

Developer Guide (May 2017)

Version 1.1

CONTENTS

ANNOUNCEMENT	3
SOLUTION OVERVIEW	4
REQUIREMENT	5
PRE-REQUISITE	6
CREATE IPARAMETER DATABASE	6
CREATE IPARAMETER TABLES	6
CREATE IPARAMETER STORED PROCEDURES	8
CREATE IPARAMETER TABLE FUNCTION.....	13
GETTING STARTED.....	14
CREATE IPARAMETER DATA SOURCE.....	15
CREATE IPARAMETER ESSENTIAL PARAMETERS.....	16
CREATE IPARAMETER DATASETS	17
CREATE IPARAMETER CASCADING PARAMETERS.....	18
(OPTIONAL) APPLY JOIN	19
(OPTIONAL) REVISE MAIN QUERY SCRIPT	20
ATTACH SAVE PARAMETER STORED PROCEDURE	21
CREATE ROW IN REPORT & REPORTPARAMDETAIL	22
OPTIMIZE VIEWING EXPERIENCE.....	23

CiQuest Lab. makes no warranties and representations with respect to the contents of this document.

The contents of this user guide have been tested in limited samples with success as of its date of publication. Changes to this document will be updated on CiQuest Lab. website.

We welcome user comments/feedback, and reserve the right to revise, make changes to the project, software or this document at any time without further notice.

CiQuest Lab. is not responsible for any data loss. You should always test this solution in a testing environment rather than implement this in your live environment directly.

iParameter SSRS is a CC BY-NC 4.0 (<https://creativecommons.org/licenses/by-nc/4.0/>) license project. To clarify, you can use iParameter SSRS for free (apply it in your report), in your work, but you can't sell this project itself to anyone.



CiQuest Lab.

Website	http://www.ciquest.com
Facebook	https://twitter.com/CiQuestLab
Twitter	https://twitter.com/CiQuestLab
Email	ciquestlab@gmail.com

Solution Overview

iParameter SSRS enables SSRS developer to create report with parameter saving feature: create, update and delete parameter. It's a native SQL based solution that you can apply to most your SSRS reports and save tremendous time for report viewers.

Report viewer can easily create preferred parameter set, name it and save it; if the parameter is no longer needed, user can simply delete it. Also, existing parameter set can be applied to subscription.

Requirement

To implement iParameter SSRS, you need to meet the following requirement,

Minimum requirement

Microsoft SQL Server 2008 R2 or above for Reporting Service

sysadmin permission to the database server for create iParameter database

(Optional) IE^{*1}10, 11 with CiQuest iParameter Add-ons or Google Chrome^{*2} with CiQuest iParameter extension installed.

Recommended

Microsoft SQL Server 2016 Express Edition with Advanced Services

Active Directory for iParameter user control

(Optional) For best report viewing experience, Google Chrome^{*2} with iParameter extension is highly recommended.

^{*1} IE10, 11 with CiQuest iParameter Add-ons is only compatible with SSRS 2008 R2

^{*2} Google Chrome with CiQuest iParameter extension is compatible with all iParameter SSRS reports on SSRS 2008 R2 or above

Pre-requisite

In order to apply iParameter SSRS to SSRS report, you will need to create the following components,

1. 1 database: (CiQuest)
2. 2 tables: (Report, ReportParamDetail)
3. 2 stored procedures
4. 1 table function

Note: Rename above components (database, table, procedure or table function) is highly NOT recommended.

Create iParameter Database

Create a database – CiQuest (Name change not recommended). Set recovery mode according to your preference, full recovery by default.

Create iParameter Tables

iParameter SSRS requires 2 essential tables to work properly, run below script to create,

Report

```
CREATE TABLE [dbo].[Report](
    [ReportID] [uniqueidentifier] NOT NULL,
    [ReportName] [nvarchar](50) NOT NULL,
    [ReportStatus] [tinyint] NOT NULL,
    CONSTRAINT [PK_Report] PRIMARY KEY CLUSTERED
(
    [ReportID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

ReportParamDetail

```

CREATE TABLE [dbo].[ReportParamDetail](
    [ParamPDID] [int] IDENTITY(1,1) NOT NULL,
    [ReportID] [uniqueidentifier] NOT NULL,
    [UserLogin] [nvarchar](50) NULL,
    [ParamSetName] [nvarchar](50) NOT NULL,
    [ParamSetDesc] [nvarchar](100) NULL,
    [Order] [tinyint] NULL,
    [ParamValue1] [nvarchar](max) NULL,
    [ParamValue2] [nvarchar](max) NULL,
    [ParamValue3] [nvarchar](max) NULL,
    [ParamValue4] [nvarchar](max) NULL,
    [ParamValue5] [nvarchar](max) NULL,
    [ParamValue6] [nvarchar](max) NULL,
    [ParamValue7] [nvarchar](max) NULL,
    [ParamValue8] [nvarchar](max) NULL,
    [ParamValue9] [nvarchar](max) NULL,
    [ParamValue10] [nvarchar](max) NULL,
    [ParamValue11] [nvarchar](max) NULL,
    [ParamValue12] [nvarchar](max) NULL,
    [ParamValue13] [nvarchar](max) NULL,
    [ParamValue14] [nvarchar](max) NULL,
    [ParamValue15] [nvarchar](max) NULL,
    [IsDefault] [tinyint] NOT NULL,
    [LastModified] [datetime] NOT NULL,
    CONSTRAINT [PK_ReportParamDetail] PRIMARY KEY CLUSTERED
(
    [ParamPDID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

ALTER TABLE [dbo].[ReportParamDetail] ADD CONSTRAINT
[DF_ReportParamDetail_IsDefault] DEFAULT ((0)) FOR [IsDefault]
GO

ALTER TABLE [dbo].[ReportParamDetail] ADD CONSTRAINT
[DF_ReportParamDetail_LastModified] DEFAULT (getdate()) FOR [LastModified]
GO

ALTER TABLE [dbo].[ReportParamDetail] WITH CHECK ADD CONSTRAINT
[FK_ReportParamDetail_Report] FOREIGN KEY([ReportID])
REFERENCES [dbo].[Report] ([ReportID])
GO

ALTER TABLE [dbo].[ReportParamDetail] CHECK CONSTRAINT
[FK_ReportParamDetail_Report]
GO

```

Create iParameter Stored Procedures

Save Report Parameters

```
CREATE procedure [dbo].[sp_Save_ReportParamDetail]
(
    @SaveParameter          int,      --0:View, 1:Create, 2:Save, 3:Delete
    @Parameter              int,      --ReportPDID in actual table
    @ReportID               uniqueidentifier,
    @UserLogin              nvarchar(50),
    @NewName                nvarchar(50), --ParamSetName in actual table
    @ParamSetDesc           nvarchar(100),
    @ParamValue1_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue1            nvarchar(max),
    @ParamValue2_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue2            nvarchar(max),
    @ParamValue3_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue3            nvarchar(max),
    @ParamValue4_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue4            nvarchar(max),
    @ParamValue5_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue5            nvarchar(max),
    @ParamValue6_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue6            nvarchar(max),
    @ParamValue7_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue7            nvarchar(max),
    @ParamValue8_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue8            nvarchar(max),
    @ParamValue9_ODT        tinyint, --0:Text, 1:DateTime
    @ParamValue9            nvarchar(max),
    @ParamValue10_ODT       tinyint, --0:Text, 1:DateTime
    @ParamValue10           nvarchar(max),
    @ParamValue11_ODT       tinyint, --0:Text, 1:DateTime
    @ParamValue11           nvarchar(max),
    @ParamValue12_ODT       tinyint, --0:Text, 1:DateTime
    @ParamValue12           nvarchar(max),
    @ParamValue13_ODT       tinyint, --0:Text, 1:DateTime
    @ParamValue13           nvarchar(max),
    @ParamValue14_ODT       tinyint, --0:Text, 1:DateTime
    @ParamValue14           nvarchar(max),
    @ParamValue15_ODT       tinyint, --0:Text, 1:DateTime
    @ParamValue15           nvarchar(max),
    @IsDefault              tinyint = 0 --0:No, 1:Yes
)
as
begin
    if (@SaveParameter = 1)
    begin
        if (@NewName is null or @NewName = '')
        begin
            raiserror('Error: [New Parameter Name] cannot be empty while
creating new profile', 16, 1)
        end
        else
        begin
            insert into CiQuest.dbo.ReportParamDetail (ReportID, UserLogin,
ParamSetName, ParamValue1, ParamValue2, ParamValue3, ParamValue4, ParamValue5,
```



```

ParamValue6, ParamValue7, ParamValue8, ParamValue9, ParamValue10, ParamValue11,
ParamValue12, ParamValue13, ParamValue14, ParamValue15, IsDefault, LastModified)
    select @ReportID, @UserLogin, @NewName,
           case when @ParamValue1_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue1), 23) else @ParamValue1 end,
           case when @ParamValue2_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue2), 23) else @ParamValue2 end,
           case when @ParamValue3_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue3), 23) else @ParamValue3 end,
           case when @ParamValue4_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue4), 23) else @ParamValue4 end,
           case when @ParamValue5_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue5), 23) else @ParamValue5 end,
           case when @ParamValue6_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue6), 23) else @ParamValue6 end,
           case when @ParamValue7_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue7), 23) else @ParamValue7 end,
           case when @ParamValue8_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue8), 23) else @ParamValue8 end,
           case when @ParamValue9_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue9), 23) else @ParamValue9 end,
           case when @ParamValue10_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue10), 23) else @ParamValue10 end,
           case when @ParamValue11_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue11), 23) else @ParamValue11 end,
           case when @ParamValue12_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue12), 23) else @ParamValue12 end,
           case when @ParamValue13_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue13), 23) else @ParamValue13 end,
           case when @ParamValue14_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue14), 23) else @ParamValue14 end,
           case when @ParamValue15_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue15), 23) else @ParamValue15 end,
           @IsDefault, GetDate()
    end
    else if (@SaveParameter = 2)      --update
    begin
        declare @IsParamExist int
        set @IsParamExist = (select COUNT(ParamPDID) from
CiQuest.dbo.ReportParamDetail where ParamPDID = @Parameter)

        if (@IsParamExist > 0)
        begin
            update CiQuest.dbo.ReportParamDetail
            set
                ParamSetName = case when @NewName = null or @NewName =
'' then ParamSetName else @NewName end,
                ParamValue1 = case when @ParamValue1_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue1), 23) else @ParamValue1 end,
                ParamValue2 = case when @ParamValue2_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue2), 23) else @ParamValue2 end,
                ParamValue3 = case when @ParamValue3_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue3), 23) else @ParamValue3 end,
                ParamValue4 = case when @ParamValue4_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue4), 23) else @ParamValue4 end,
                ParamValue5 = case when @ParamValue5_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue5), 23) else @ParamValue5 end,

```

```

        ParamValue6 = case when @ParamValue6_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue6), 23) else @ParamValue6 end,
        ParamValue7 = case when @ParamValue7_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue7), 23) else @ParamValue7 end,
        ParamValue8 = case when @ParamValue8_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue8), 23) else @ParamValue8 end,
        ParamValue9 = case when @ParamValue9_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue9), 23) else @ParamValue9 end,
        ParamValue10 = case when @ParamValue10_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue10), 23) else @ParamValue10 end,
        ParamValue11 = case when @ParamValue11_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue11), 23) else @ParamValue11 end,
        ParamValue12 = case when @ParamValue12_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue12), 23) else @ParamValue12 end,
        ParamValue13 = case when @ParamValue13_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue13), 23) else @ParamValue13 end,
        ParamValue14 = case when @ParamValue14_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue14), 23) else @ParamValue14 end,
        ParamValue15 = case when @ParamValue15_ODT = 1 then
convert(nvarchar(10), convert(datetime, @ParamValue15), 23) else @ParamValue15 end,
        IsDefault = @IsDefault, LastModified = getDate()

        where ParamPDID = @Parameter
            and IsDefault = 0

    end
    else
    begin
        raiserror('Error: The profile you are trying to save does not
exist!', 16, 1)
    end

    end
    else if (@SaveParameter = 3)        --delete
    begin
        set @IsDefault = (select IsDefault from
CiQuest.dbo.ReportParamDetail where ParamPDID = @Parameter)
        if (@IsDefault != 1)
        begin
            delete from CiQuest.dbo.ReportParamDetail where ParamPDID =
@Parameter and IsDefault = 0
        end
    end
end
GO

```

Get Report Parameters

```
CREATE procedure [dbo].[sp_Get_ReportParamDetailbyMode]
(
    @Mode                int,    --0: Default, 1:Valid, 2:Selected
    @ReportID            uniqueidentifier,
    @Parameter           int = null,
    @UserLogin           nvarchar(50),
    @ParamValue1_ODT     int,    --0:Text, 1:DateTime
    @ParamValue2_ODT     int,    --0:Text, 1:DateTime
    @ParamValue3_ODT     int,    --0:Text, 1:DateTime
    @ParamValue4_ODT     int,    --0:Text, 1:DateTime
    @ParamValue5_ODT     int,    --0:Text, 1:DateTime
    @ParamValue6_ODT     int,    --0:Text, 1:DateTime
    @ParamValue7_ODT     int,    --0:Text, 1:DateTime
    @ParamValue8_ODT     int,    --0:Text, 1:DateTime
    @ParamValue9_ODT     int,    --0:Text, 1:DateTime
    @ParamValue10_ODT    int,    --0:Text, 1:DateTime
    @ParamValue11_ODT    int,    --0:Text, 1:DateTime
    @ParamValue12_ODT    int,    --0:Text, 1:DateTime
    @ParamValue13_ODT    int,    --0:Text, 1:DateTime
    @ParamValue14_ODT    int,    --0:Text, 1:DateTime
    @ParamValue15_ODT    int     --0:Text, 1:DateTime
)
as
begin
    declare @SQL nvarchar(max), @ParamDefinition nvarchar(200)
    set @SQL = N'select ParamPDID, ReportID, UserLogin, ParamSetName, '

    if (@ParamValue1_ODT = 0) begin set @SQL = @SQL + ' ParamValue1, ' end
    else begin set @SQL = @SQL + 'convert(datetime, ParamValue1) as ParamValue1, ' end
' end

    if (@ParamValue2_ODT = 0) begin set @SQL = @SQL + ' ParamValue2, ' end
    else begin set @SQL = @SQL + 'convert(datetime, ParamValue2) as ParamValue2, ' end
' end

    if (@ParamValue3_ODT = 0) begin set @SQL = @SQL + ' ParamValue3, ' end
    else begin set @SQL = @SQL + 'convert(datetime, ParamValue3) as ParamValue3, ' end
' end

    if (@ParamValue4_ODT = 0) begin set @SQL = @SQL + ' ParamValue4, ' end
    else begin set @SQL = @SQL + 'convert(datetime, ParamValue4) as ParamValue4, ' end
' end

    if (@ParamValue5_ODT = 0) begin set @SQL = @SQL + ' ParamValue5, ' end
    else begin set @SQL = @SQL + 'convert(datetime, ParamValue5) as ParamValue5, ' end
' end

    if (@ParamValue6_ODT = 0) begin set @SQL = @SQL + ' ParamValue6, ' end
    else begin set @SQL = @SQL + 'convert(datetime, ParamValue6) as ParamValue6, ' end
' end

    if (@ParamValue7_ODT = 0) begin set @SQL = @SQL + ' ParamValue7, ' end
    else begin set @SQL = @SQL + 'convert(datetime, ParamValue7) as ParamValue7, ' end
' end
```

```

        if (@ParamValue8_ODT = 0) begin set @SQL = @SQL + ' ParamValue8, ' end
        else begin set @SQL = @SQL + 'convert(datetime, ParamValue8) as ParamValue8,
' end

        if (@ParamValue9_ODT = 0) begin set @SQL = @SQL + ' ParamValue9, ' end
        else begin set @SQL = @SQL + 'convert(datetime, ParamValue9) as ParamValue9,
' end

        if (@ParamValue10_ODT = 0) begin set @SQL = @SQL + ' ParamValue10, ' end
        else begin set @SQL = @SQL + 'convert(datetime, ParamValue10) as
ParamValue10, ' end

        if (@ParamValue11_ODT = 0) begin set @SQL = @SQL + ' ParamValue11, ' end
        else begin set @SQL = @SQL + 'convert(datetime, ParamValue11) as
ParamValue11, ' end

        if (@ParamValue12_ODT = 0) begin set @SQL = @SQL + ' ParamValue12, ' end
        else begin set @SQL = @SQL + 'convert(datetime, ParamValue12) as
ParamValue12, ' end

        if (@ParamValue13_ODT = 0) begin set @SQL = @SQL + ' ParamValue13, ' end
        else begin set @SQL = @SQL + 'convert(datetime, ParamValue13) as
ParamValue13, ' end

        if (@ParamValue14_ODT = 0) begin set @SQL = @SQL + ' ParamValue14, ' end
        else begin set @SQL = @SQL + 'convert(datetime, ParamValue14) as
ParamValue14, ' end

        if (@ParamValue15_ODT = 0) begin set @SQL = @SQL + ' ParamValue15, ' end
        else begin set @SQL = @SQL + 'convert(datetime, ParamValue15) as
ParamValue15, ' end

        set @SQL = @SQL + 'IsDefault, LastModified from ReportParamDetail where
ReportID = @Int_ReportID '
        if (@Mode = 0) --Default
        begin
            set @SQL = @SQL + 'and IsDefault = 1'
        end
        else if (@Mode = 1) --Valid
        begin
            set @SQL = @SQL + ' and (UserLogin = @Int_UserLogin or UserLogin is
null) order by IsDefault desc, ParamSetName'
        end
        else --@Mode = 2 --Selected
        begin
            set @SQL = @SQL + ' and ParamPDID = @Int_Parameter'
        end
        end

        set @ParamDefinition = N'@Int_ReportID uniqueidentifier, @Int_UserLogin
nvarchar(50), @Int_Parameter int'

EXECUTE sp_executesql @SQL, @ParamDefinition, @Int_ReportID = @ReportID,
@Int_UserLogin = @UserLogin, @Int_Parameter = @Parameter
end
GO

```

Create iParameter Table Function

```
CREATE FUNCTION [dbo].[fn_charlist_to_table]
(
    @list ntext,
    @delimiter nchar(1) = N','
)
RETURNS @tbl TABLE
(
    listpos int IDENTITY(1, 1) NOT NULL,
    str varchar(4000),
    nstr nvarchar(2000)
)
AS
BEGIN
    DECLARE @pos int, @textpos int, @chunklen smallint, @tmpstr nvarchar(4000),
    @leftover nvarchar(4000), @tmpval nvarchar(4000)

    SET @textpos = 1
    SET @leftover = ''

    WHILE (@textpos <= datalength(@list) / 2)
    BEGIN
        SET @chunklen = 4000 - datalength(@leftover) / 2
        SET @tmpstr = @leftover + substring(@list, @textpos, @chunklen)
        SET @textpos = @textpos + @chunklen
        SET @pos = charindex(@delimiter, @tmpstr)

        WHILE (@pos > 0)
        BEGIN
            SET @tmpval = ltrim(rtrim(left(@tmpstr, @pos - 1)))
            INSERT @tbl (str, nstr) VALUES (@tmpval, @tmpval)
            SET @tmpstr = substring(@tmpstr, @pos + 1, len(@tmpstr))
            SET @pos = charindex(@delimiter, @tmpstr)
        END

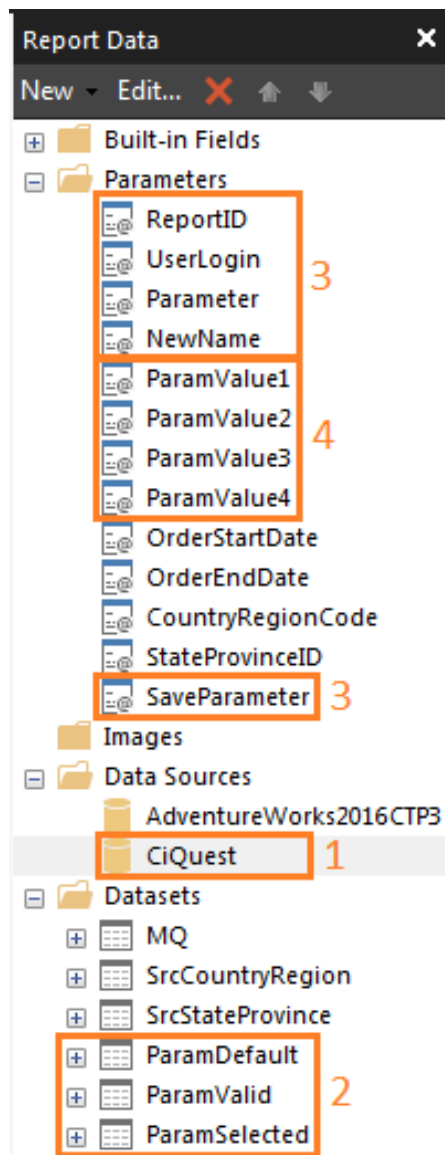
        SET @leftover = @tmpstr
    END

    INSERT @tbl (str, nstr) VALUES (ltrim(rtrim(@leftover)),
    ltrim(rtrim(@leftover)))

    RETURN
END
GO
```

Getting Started

To implement iParameter SSRS, you will need to follow below steps,



1. Create iParameter Data Source
2. Create iParameter Essential Parameters
3. Create iParameter Cascading Parameters
4. Create iParameter Datasets
5. (Optional) Apply Join
6. (Optional) Revise Main Query Script – Replace MV parameter with table function
7. Append Save Report Parameter Stored Procedure
8. Create row in Report and ReportParamDetail

Section 1: Create iParameter Data Source

Create a new Data Source, name: CiQuest, and point it to CiQuest database. This is the database where all your report parameters will be saved.

Shared Data Source is recommended.

Section 2: Create iParameter Essential Parameters

Create 5 iParameter essential parameters,

1. Name ReportID
Data type Text
Visibility Hidden
Available Values None
Default Values Specify values - create an UniqueID and paste it
2. Name UserLogin
Data type Text
Visibility Hidden
Available Values None
Default Values =User!UserID
3. Name Parameter
Data type Integer [Change to below after dataset created]
Visibility Visible
Available Values None

→

Dataset ParamValid
 Value field ParamPDID
 Label field ParamSetName

→

Get values from a query
 Dataset ParamDefault
 Value field ParamPDID

Default Values None
4. Name NewName
Data type Text
Allow blank value ("") Check
Visibility Visible
Available Values None
Default Values None
5. Name SaveParameter
Data type Integer
Visibility Visible
Available Values Specify values
See Figure

Label	Value
View Report	0
Save as New (Refresh Browser Requi	1
Save	2
Delete	3

Default Values: Specify values. Value: 0

Note: The first 4 essential parameters MUST be stay on top place.

Section 3: Create iParameter Datasets

Create 3 iParameter Datasets

1. Name ParamDefault
Data source CiQuest
Query type Stored Procedure
Procedure Name sp_Get_ReportParamDetailbyMode
Name Value
Parameters @Mode 0
 @ReportID =Parameters!ReportID.Value
 @Parameter -1
 @UserLogin = Parameters!UserLogin.Value
Apply 0 to all ODT parameters, @ParamValue1_ODT to @ParamValue15_ODT

2. Name: ParamValid
Data source CiQuest
Query type Stored Procedure
Procedure Name sp_Get_ReportParamDetailbyMode
Name Value
Parameters @Mode 1
 @ReportID =Parameters!ReportID.Value
 @Parameter -1
 @UserLogin = Parameters!UserLogin.Value
Apply 0 to all ODT parameters, @ParamValue1_ODT to @ParamValue15_ODT

3. Name: ParamSelected
Data source CiQuest
Query type Stored Procedure
Procedure Name sp_Get_ReportParamDetailbyMode
Name Value
Parameters @Mode 2
 @ReportID =Parameters!ReportID.Value
 @Parameter =Parameters!Parameter.Value
 @UserLogin = Parameters!UserLogin.Value
Apply 0 to all ODT parameters, @ParamValue1_ODT to @ParamValue15_ODT

Section 4: Create iParameter Cascading Parameters

The cascading parameter is used to bring all your saved parameters back from the parameter dropdown you selected. By design, you will make the decision on how many parameter(s) will be saved. If your report has 8 parameters in total, and you wish to save 4 of them, you will need to create 4 cascading parameters, they will be,

ParamValue1, ParamValue2, ParamValue3 and ParamValue4

Each cascading parameter should be created like below,

If your parameter data type is "Date/Time", create one cascading parameter like below,

Name	ParamValue1 (Change 1 to other number in order)
Data type	Date/Time
Allow multiple values	Check (Always)
Visibility	Visible
Available Values	Get values from a query
- Dataset	ParamSelected
- Value field	ParamValue1 (Change 1 to other number in order)
- Label field	ParamValue1 (Change 1 to other number in order)
Default Values	ParamSelected
- Value field	ParamValue1 (Change 1 to other number in order)

And change your original parameter default values to

Specify values =Parameters!Param1.Value

Otherwise, create one like below,

Name	ParamValue1 (Change 1 to other number in order)
Data type	Text
Allow multiple values	Check (Always)
Visibility	Visible
Available Values	Get values from a query
- Dataset	ParamSelected
- Value field	ParamValue1 (Change 1 to other number in order)
- Label field	ParamValue1 (Change 1 to other number in order)
Default Values	ParamSelected
- Value field	ParamValue1 (Change 1 to other number in order)

And change your original parameter default values to

If original parameter is single value

Specify values =Parameters!Param1.Value

If original parameter allow multiple values

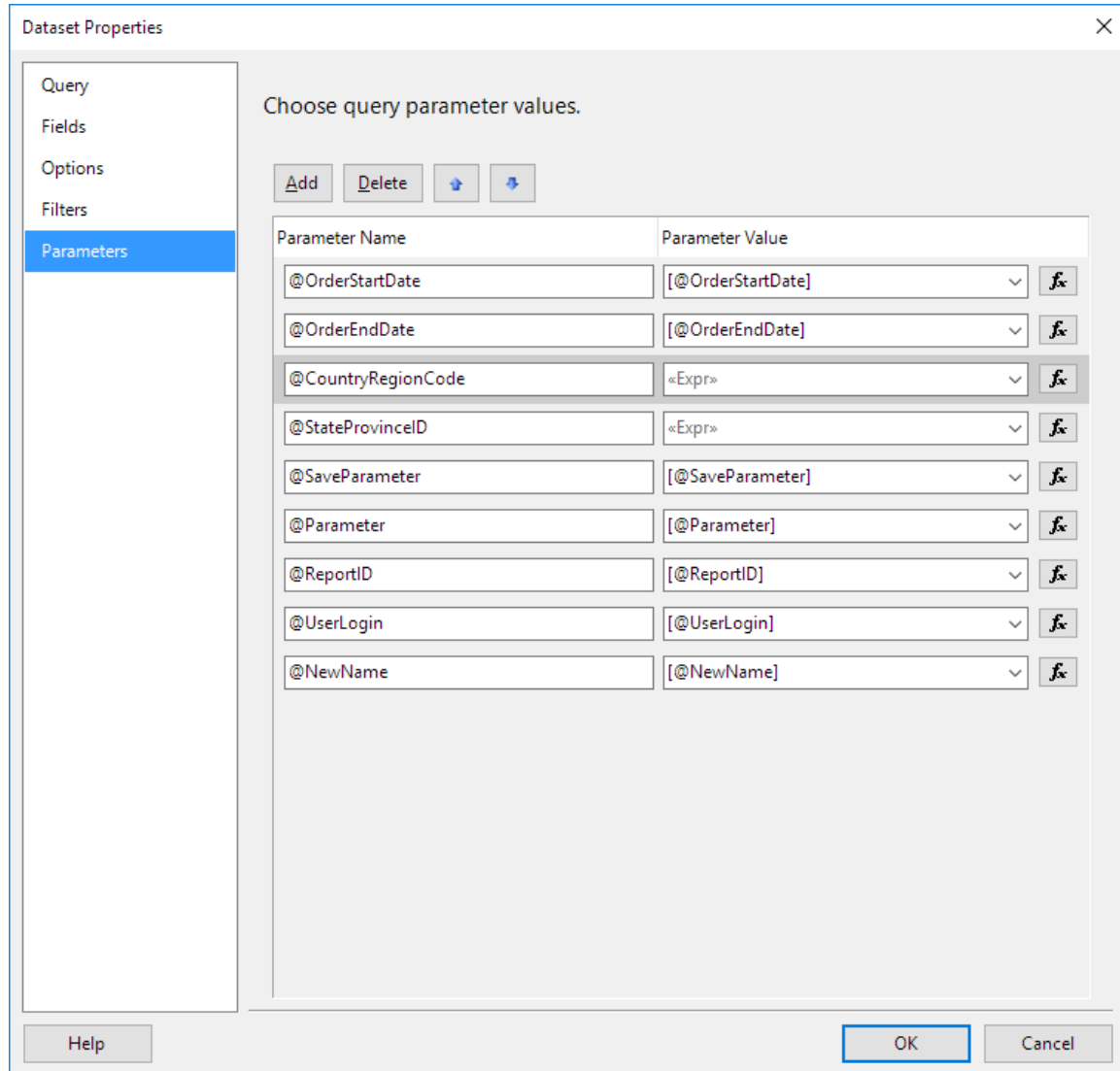
Specify values =Split(CStr(Parameters!Param1.Value(0)),",")

Note: The cascading parameter MUST be in prior place than your original parameter.

Note: Keep all cascading parameters together, between NewName and your first one.

Section 5: (Optional) Apply Join

The Join function is only used when you have wish to save parameters with multiple values to iParameter. This requires you to change your main query (main dataset for report output) Parameters. See below,



The screenshot shows the 'Dataset Properties' dialog box with the 'Parameters' tab selected. The 'Choose query parameter values' section contains a table with two columns: 'Parameter Name' and 'Parameter Value'. The parameters listed are:

Parameter Name	Parameter Value
@OrderStartDate	[@OrderStartDate]
@OrderEndDate	[@OrderEndDate]
@CountryRegionCode	«Expr»
@StateProvinceID	«Expr»
@SaveParameter	[@SaveParameter]
@Parameter	[@Parameter]
@ReportID	[@ReportID]
@UserLogin	[@UserLogin]
@NewName	[@NewName]

The '@CountryRegionCode' parameter is highlighted, and its value is set to an expression '«Expr»'. The 'OK' button is highlighted.

Figure 5-1. Parameters in Main Query Dataset

For example, in figure 5-1, the @CountryRegionCode allows multiple values. The original value would be =Parameters!CountryRegionCode.Value, you will need to change this to

Expression

Set expression for: Value

```
=Join(Parameters!CountryRegionCode.Value, ", ")
```

Section 6: (Optional) Revise Main Query Script

Also, in your main query dataset, you will need to apply change to your parameter(s) with multiple values from

@ParameterName

to

select [str] from CiQuest.dbo.fn_charlist_to_table(@CountryRegionCode,Default)

These parameters are usually used in “IN” clause, like below,

Query:

```
select cr.CountryRegionCode, cr.Name, sp.StateProvinceCode, sp.Name as StateName, sum
(SubTotal) as SubTotalSum, sum(TaxAmt) as TaxAmtSum, sum(Freight) as FreightSum, sum
(TotalDue) as TotalDueSum
from Sales.SalesOrderHeader so
    join Person.[Address] a on so.ShipToAddressID = a.AddressID
    join Person.StateProvince sp on a.StateProvinceID = sp.StateProvinceID
    join Person.CountryRegion cr on sp.CountryRegionCode = cr.CountryRegionCode
where so.OrderDate between @OrderStartDate and @OrderEndDate
    and cr.CountryRegionCode in (@CountryRegionCode)
    and sp.StateProvinceID in (@StateProvinceID)
group by cr.CountryRegionCode, cr.Name, sp.StateProvinceCode, sp.Name

----Embed Below Script for ReportParamDetail----
```

Figure 6-1. Original Query Script

Query:

```
select cr.CountryRegionCode, cr.Name, sp.StateProvinceCode, sp.Name as
StateName, sum(SubTotal) as SubTotalSum, sum(TaxAmt) as TaxAmtSum, sum
(Freight) as FreightSum, sum(TotalDue) as TotalDueSum
from Sales.SalesOrderHeader so
    join Person.[Address] a on so.ShipToAddressID = a.AddressID
    join Person.StateProvince sp on a.StateProvinceID = sp.StateProvinceID
    join Person.CountryRegion cr on sp.CountryRegionCode =
cr.CountryRegionCode
where so.OrderDate between @OrderStartDate and @OrderEndDate
    and cr.CountryRegionCode in (select [str] from
CiQuest.dbo.fn_charlist_to_table(@CountryRegionCode,Default))
    and sp.StateProvinceID in (select [str] from
CiQuest.dbo.fn_charlist_to_table(@StateProvinceID,Default))
group by cr.CountryRegionCode, cr.Name, sp.StateProvinceCode, sp.Name
/* iParameter by CiQuest(R)2017 http://www.ciqest.com */exec
```

Figure 6-2. Revised Query Script

Note: Section 5 & 6 should ONLY apply to dataset that is in charge of save report parameter stored procedure.

Section 7: Attach Save Parameter Stored Procedure

Embed the save report parameters stored procedure to your main dataset. Like below,

----Embed Below Script for ReportParamDetail----

```
exec CiQuest.dbo.sp_Save_ReportParamDetail @SaveParameter, @Parameter,  
@ReportID, @UserLogin, @NewName, null, 1, @OrderStartDate, 1, @OrderEndDate, 0,  
@CountryRegionCode, 0, @StateProvinceID, 0, null, 0, null, 0, null, 0, null, 0,  
null, 0, null, 0, null, 0, null, 0, null, 0, null, 0
```

Save Report Parameter stored procedure will save up to 15 parameters. Each parameter will come with a data type ID while saving. You will need to tell the stored procedure what's the data type ID will be,

- 0 All Non-date/time-single value parameter
- 1 Date/Time and single parameter

Save Report Parameter Stored Procedure takes the input in below,

@SaveParameter	int
@Parameter	int
@ReportID	uniqueidentifier
@UserLogin	nvarchar(50)
@NewName	nvarchar(50)
@ParamSetDesc	nvarchar(100), NULL (Not in use)
@ParamValue1_ODT	tinyint
@ParamValue1	nvarchar(max)
...	
@ParamValue15_ODT	tinyint
@ParamValue15	nvarchar(max)
@IsDefault	tinyint

Note: Always pass 0 inside report main dataset. Every report should only have one row with IsDefault = 1. This default record is important to get your report to work properly with iParameter SSRS.

Section 8: Create row in Report and ReportParamDetail

The last step to get your report to work, is to register your report to iParameter. You will need to create one row to Report table and another one row to ReportParamDetail.

Report

Go to the ReportID parameter created in Section 2, copy the default value (Unique ID). Then run below script (change below accordingly),

```
insert into Report (ReportID, ReportName, ReportStatus)
select 'YourID', 'YourReportName', 1
```

ReportParamDetail

The default entry must be entered through SSMS. Simply right click your table and choose "Edit top 200 rows",

ReportID	Same above, paste your Unique ID.
UserLogin	Leave it NULL
ParamSetName	--Select Parameter-- (No double quote)
ParamSetDesc	NULL
Order	NULL
ParamValue1	Provide a legal value corresponding to your original parameter. If date/time and single, you can use YYYY-MM-DD; If value is multiple, enter one valid value

When done, deploy your report and try it.

Optimize Viewing Experience

iParameter SSRS has Internet Explorer Add-Ons and Google Chrome Extension to hide cascading parameters for better viewing experience. In order to make the Add-Ons/Extension to work, you will need to make one change to your report.

Change your report name follows with “_” and number (of amount of cascading parameters).

For example, the sample report name is

“Order Statistics by Region”

and the sample report has 4 cascading parameters – change the report name to

“Order Statistics by Region_4”

Support Matrix

	SSRS2008R2	SSRS2012	SSRS2014	SSRS2016	SSRS2017 (vNext)
iParameter IE Add-Ons	Yes	Yes	Yes		
iParameter Chrome Ext.	Yes	Yes	Yes	Yes	Yes

You can download IE Add-ons or Google Chrome Extension from
<http://www.ciquest.com>