

AVEVA

Batch Management

Custom Report Update Guide

July 2019



© 2019 AVEVA Group plc and its subsidiaries. All rights reserved.

No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of AVEVA. No liability is assumed with respect to the use of the information contained herein.

Although precaution has been taken in the preparation of this documentation, AVEVA assumes no responsibility for errors or omissions. The information in this documentation is subject to change without notice and does not represent a commitment on the part of AVEVA. The software described in this documentation is furnished under a license agreement. This software may be used or copied only in accordance with the terms of such license agreement.

Archestra, Aquis, Avantis, Citect, DYNsIM, eDNA, EYESIM, InBatch, InduSoft, InStep, IntelaTrac, InTouch, OASyS, PIPEPHASE, PRiSM, PRO/II, PROVISION, ROMEo, SIM4ME, SimCentral, SimSci, Skelta, SmartGlance, Spiral Software, Termis, WindowMaker, WindowViewer, and Wonderware are trademarks of AVEVA and/or its subsidiaries. An extensive listing of AVEVA trademarks can be found at: <https://sw.aveva.com/legal>. All other brands may be trademarks of their respective owners.

Publication date: Wednesday, July 17, 2019

Contact Information

AVEVA Group plc
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

<https://sw.aveva.com/>

For information on how to contact sales, customer training, and technical support, see <https://sw.aveva.com/contact>.

Contents

Chapter 1	Introduction.....	5
Chapter 2	Updating Custom Report	7
	Add new Data Source: ibReports to Report	8
	Update Report Datasets	9
	Adding new Data set ReportingTimeZones to Report	12
	Adding reference to BatchTimeExtension Assembly in Report	13
	Update Report Main Dataset Expression	14
	Create new Parameter: ReportingTimeZone	16
	Update all Report Datasets expression	18
	Add new Report Code to Report	18
	Update Existing Report Code.....	20
	Other Report updates	21
	Reporting in Time Zone	23
	Sample Report with Time Zone Information	24

CHAPTER 1

Introduction

This guide provides information to assist you with the update of Batch Management Custom Reports after upgrading to Batch Management 2017 Update 1. The changes in this guide are for:

- The removal of Information Server dependencies
- Adding support for reporting in Universal Time (UTC) and in specific time zones.

It is recommended that you take a back up of the custom report before you update your custom report using this guide.

CHAPTER 2

Updating Custom Report

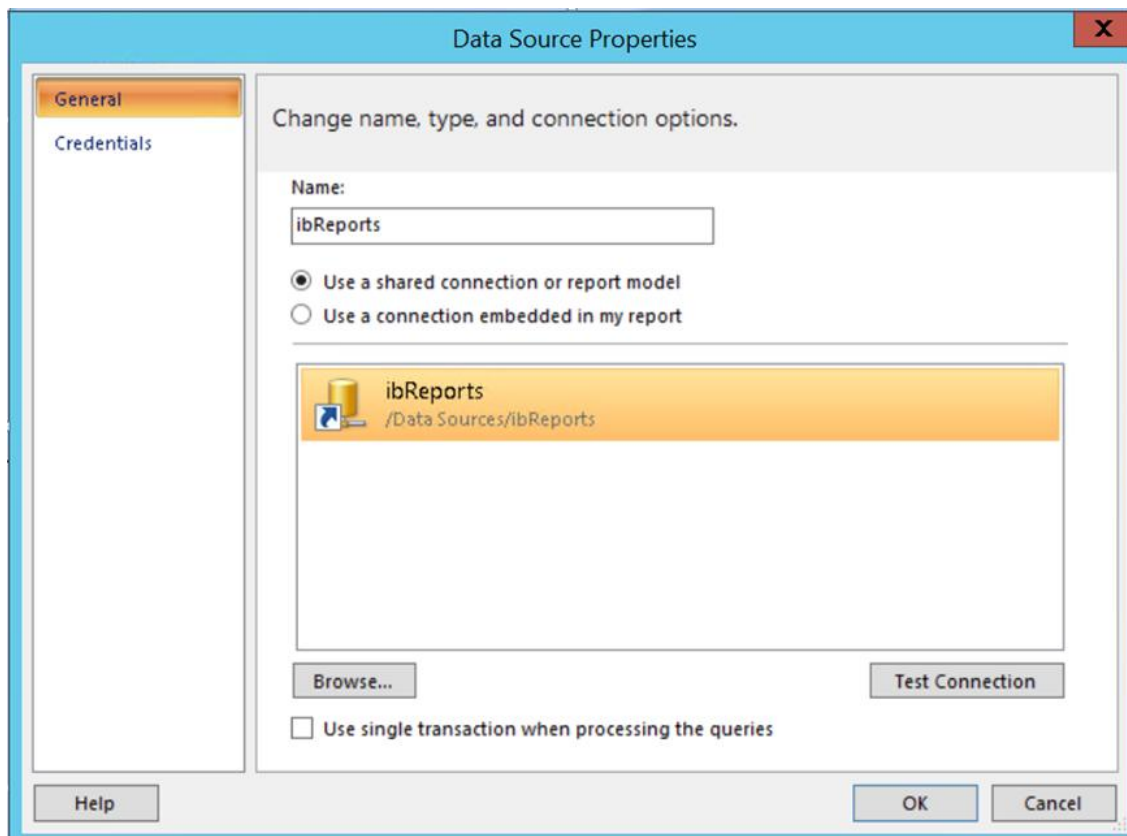
Updating the Custom Report involves the following steps:

- Add new Data Source: ibReports to Report (connection string to BatchHistory database)
- Update Report Datasets to use the new ibReports Data Source; change the Data source from “aaReports” to “ibReports”
 - Update LayoutColorCodes dataset
 - Update SystemParameters dataset
 - Update Report Datasets to use the new ibReports Data Source
- Add new Data set ReportingTimeZones to report
- Add reference to BatchTimeExtension Assembly in Report
- Update Report
 - Update main dataset expression and dataset field list
 - Add DateTime UTC column (if applicable)
- Create new Parameter: ReportingTimeZone
 - Add new Parameter: ReportingTimeZone to report, under the StartDateTime and EndDateTime parameters
 - Use ReportingTimeZones dataset to get data
- Update all Report Datasets expression
- Add new Report Code to Report
 - TimeZoneToUtc, UtcToTimeZone, GetTimeZoneDateTime, GetTimeZoneOffset, GetReportDateRange, GetReportDateColumn, GetTimeZoneSuffix, GetServerTimeZoneSuffix, and GetFixedStringFormat
- Update Existing Report Code
 - Create ValidValues Dataset
- Other Report updates
 - Update Report WhereClause to use new GetReportDateRange Function
 - Remove Deprecated WIS Report Items
 - Datasets: LayoutDurationChoices, LayoutTimeZoneInfo, Timezone (If applicable)
 - Parameters: Duration, TimeZoneId (If applicable)

In This Chapter

- Add new Data Source: ibReports to Report 8
- Update Report Datasets 9
- Adding new Data set ReportingTimeZones to Report 12
- Adding reference to BatchTimeExtension Assembly in Report 13
- Update Report Main Dataset Expression 14
- Create new Parameter: ReportingTimeZone 16
- Update all Report Datasets expression 18
- Add new Report Code to Report 18
- Update Existing Report Code 20
- Other Report updates 21
- Reporting in Time Zone 23
- Sample Report with Time Zone Information 24

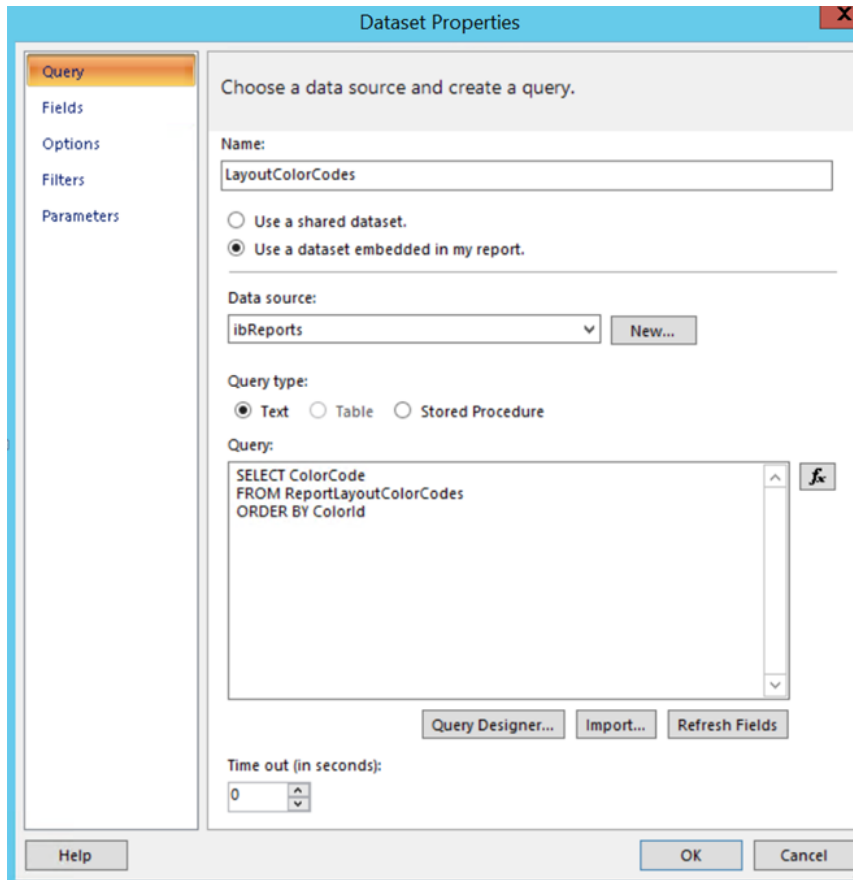
Add new Data Source: ibReports to Report



Update Report Datasets

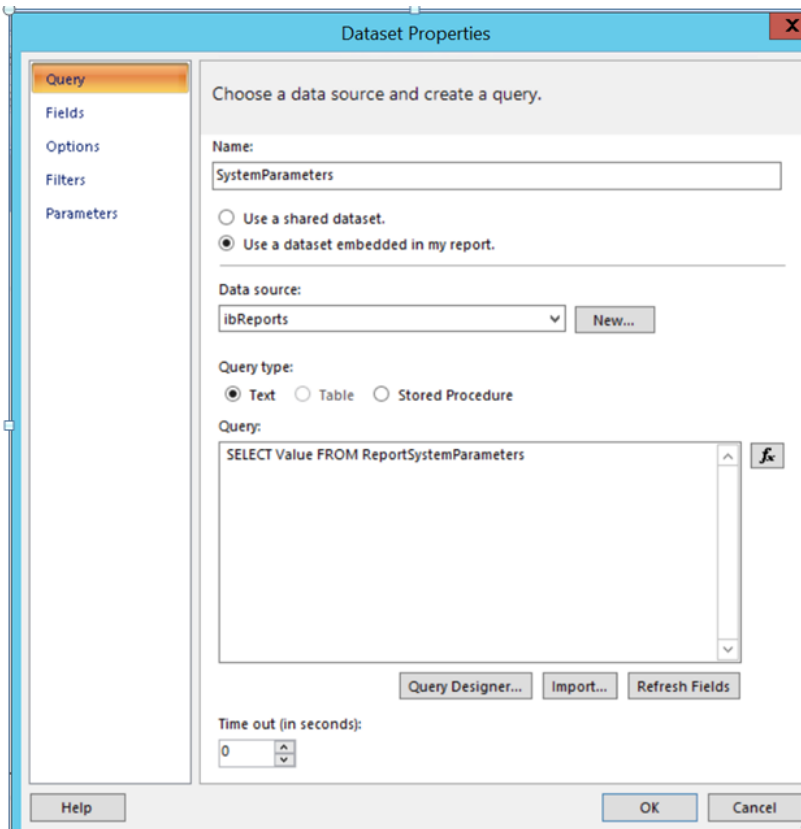
Update LayoutColorCodes Dataset

Update LayoutColorCodes dataset to use new Data Source (ibReports) and get values from new database table (ReportLayoutColorCodes).



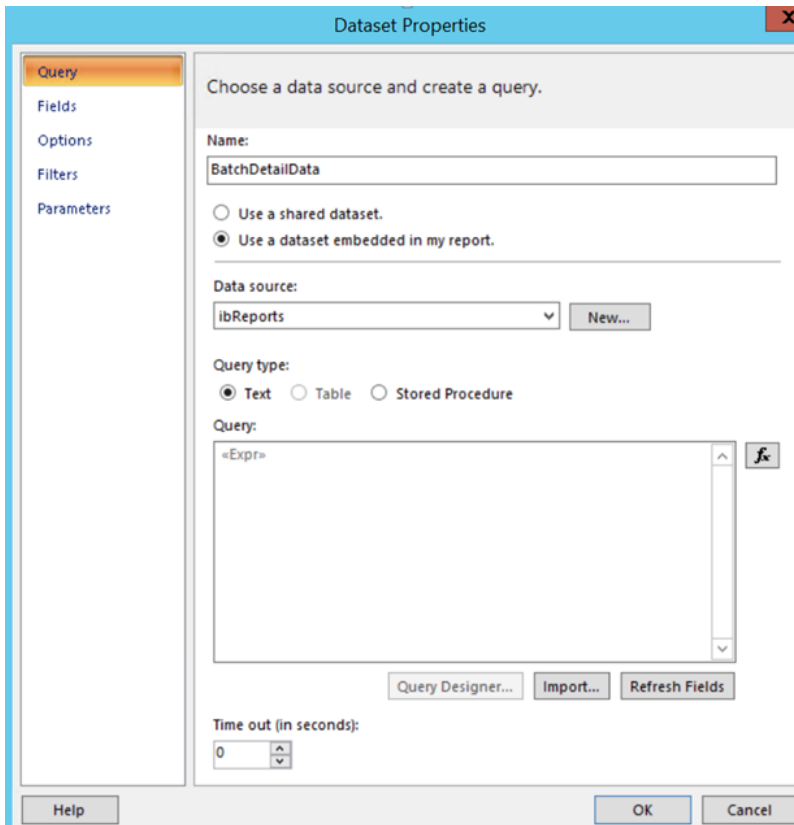
Update SystemParameters Dataset

Update SystemParameters dataset to use new Data Source (ibReports) and get values from new database table (ReportSystemParameters).

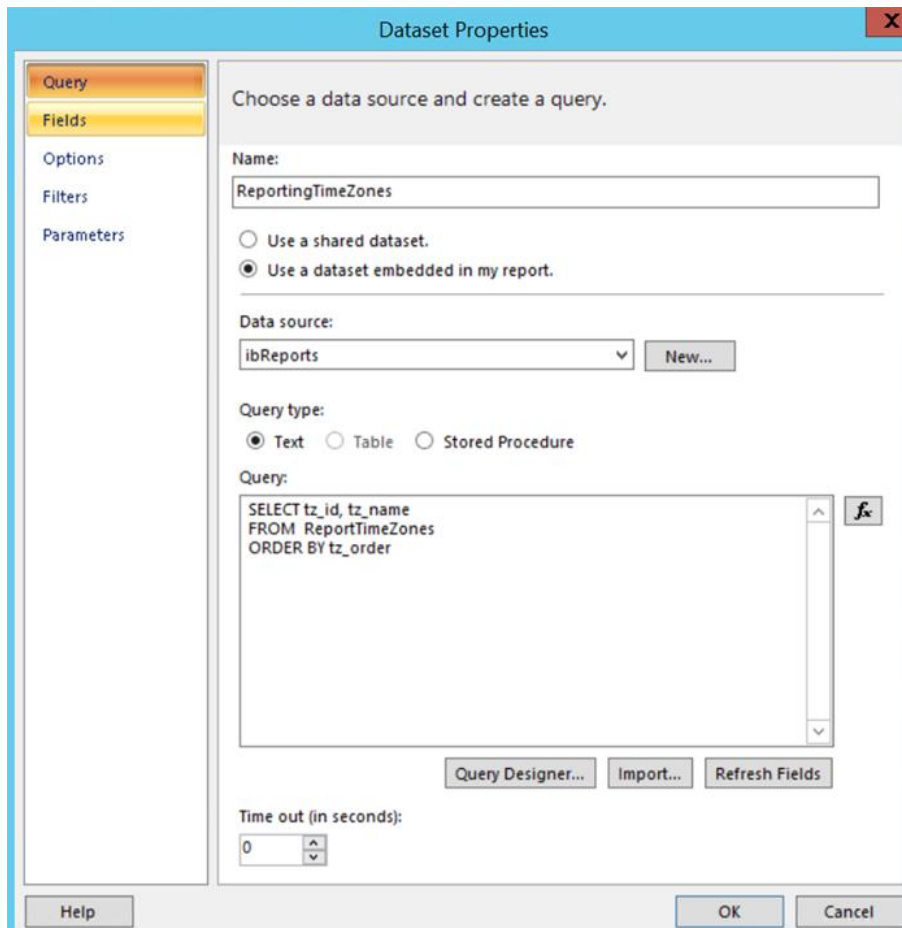


Update Report Datasets to use the new ibReports Data Source

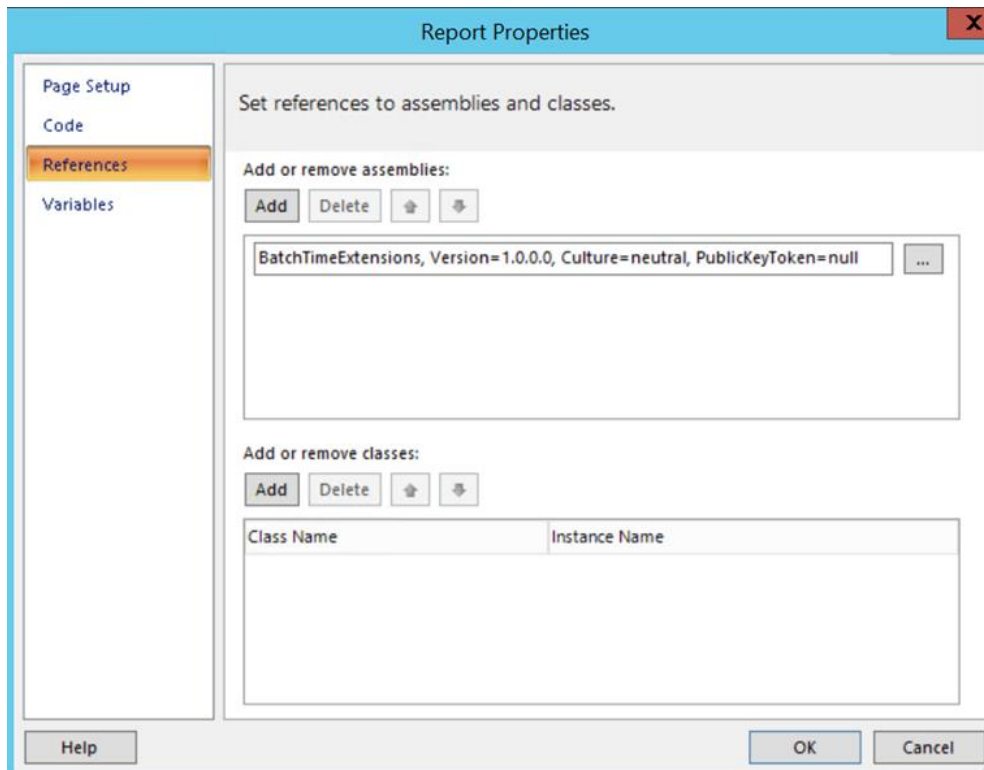
Update all other Report Datasets to use the new Data Source (ibreports). Change the Data Source from “aaReports” to “ibReports.”



Adding new Data set ReportingTimeZones to Report



Adding reference to BatchTimeExtension Assembly in Report



Location for referenced BatchTimeExtension.dll Assembly on the Report(SSRS) Server:

C:\Program Files\Microsoft SQL Server\MSRS12.MSSQLSERVER\Reporting Services\ReportServer\bin

The above path 'MSRS12.MSSQLSERVER' may vary depending on SQL Server version.

Update Report Main Dataset Expression

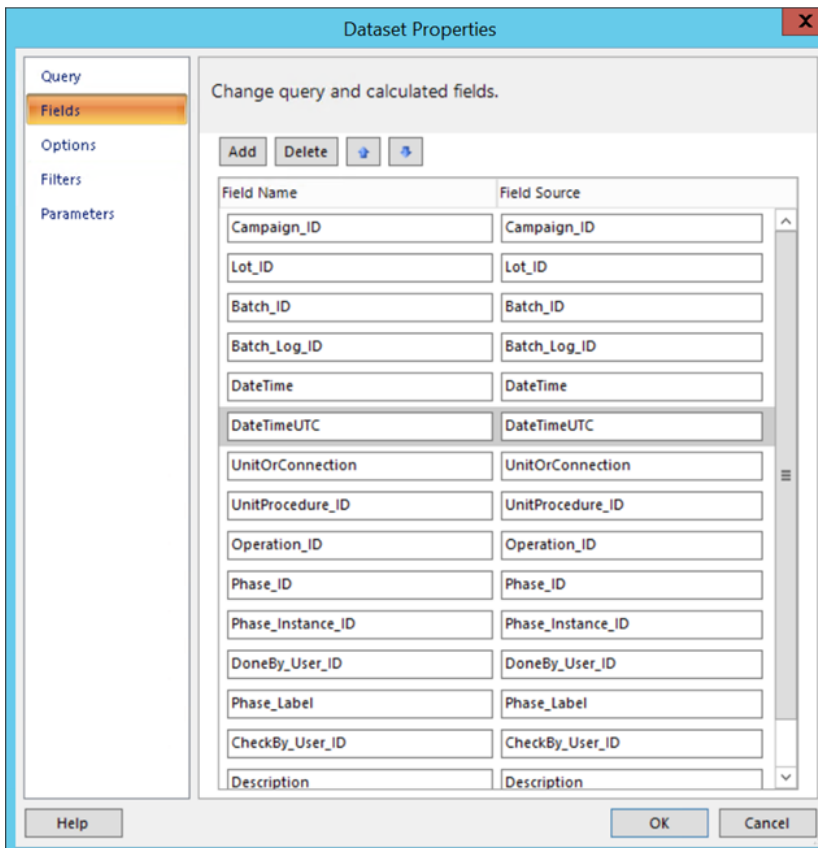
Update Report Main Dataset Expression

Update Report Main Dataset Expression (Example BatchDetailData dataset for IBBatchDetail Report).

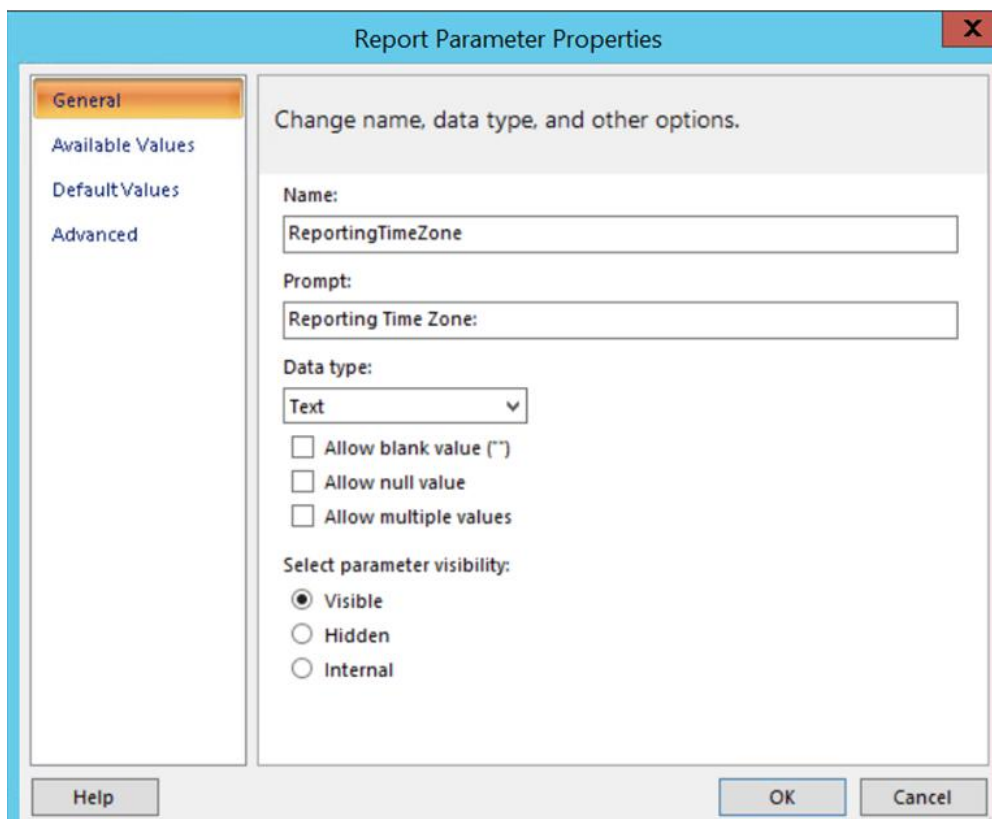
Before Update	After Update
<pre> ="SELECT BD.Campaign_ID ,BD.Lot_ID ,BD.Batch_ID ,BD.Batch_Log_ID ,BD.DateTime ,BD.UnitOrConnection ,BD.UnitProcedure_ID ,BD.Operation_ID ,BD.Phase_ID ,BD.Phase_Instance_ID ,BD.DoneBy_User_ID ,BD.Phase_Label ,BD.CheckBy_User_ID ,BD.Description ,BD.Batch_Server_Name " & "FROM IBBatchDetail BD " & Code.GetWhereClause(Parameters!WhereClause.Value) & "ORDER BY BD.Batch_Log_ID ,BD.DateTime" </pre>	<pre> ="SELECT BD.Campaign_ID ,BD.Lot_ID ,BD.Batch_ID ,BD.Batch_Log_ID ,BD.DateTime ,BD.DateTimeUTC ,BD.UnitOrConnection ,BD.UnitProcedure_ID ,BD.Operation_ID ,BD.Phase_ID ,BD.Phase_Instance_ID ,BD.DoneBy_User_ID ,BD.Phase_Label ,BD.CheckBy_User_ID ,BD.Description ,BD.Batch_Server_Name " & "FROM IBBatchDetail BD " & Code.GetWhereClause(Parameters!WhereClause.Value) & "ORDER BY BD.Batch_Log_ID ,BD.DateTime" </pre>

Update Report main dataset field list

Add UTC DateTime column to report's main dataset field list (Example BatchDetailData Dataset for IBBatchDetail Report below):



Create new Parameter: ReportingTimeZone



The screenshot shows a dialog box titled "Report Parameter Properties" with a close button (X) in the top right corner. On the left side, there is a vertical navigation pane with four tabs: "General", "Available Values" (which is highlighted in orange), "Default Values", and "Advanced". The main area of the dialog is titled "Choose the available values for this parameter." and contains the following options:

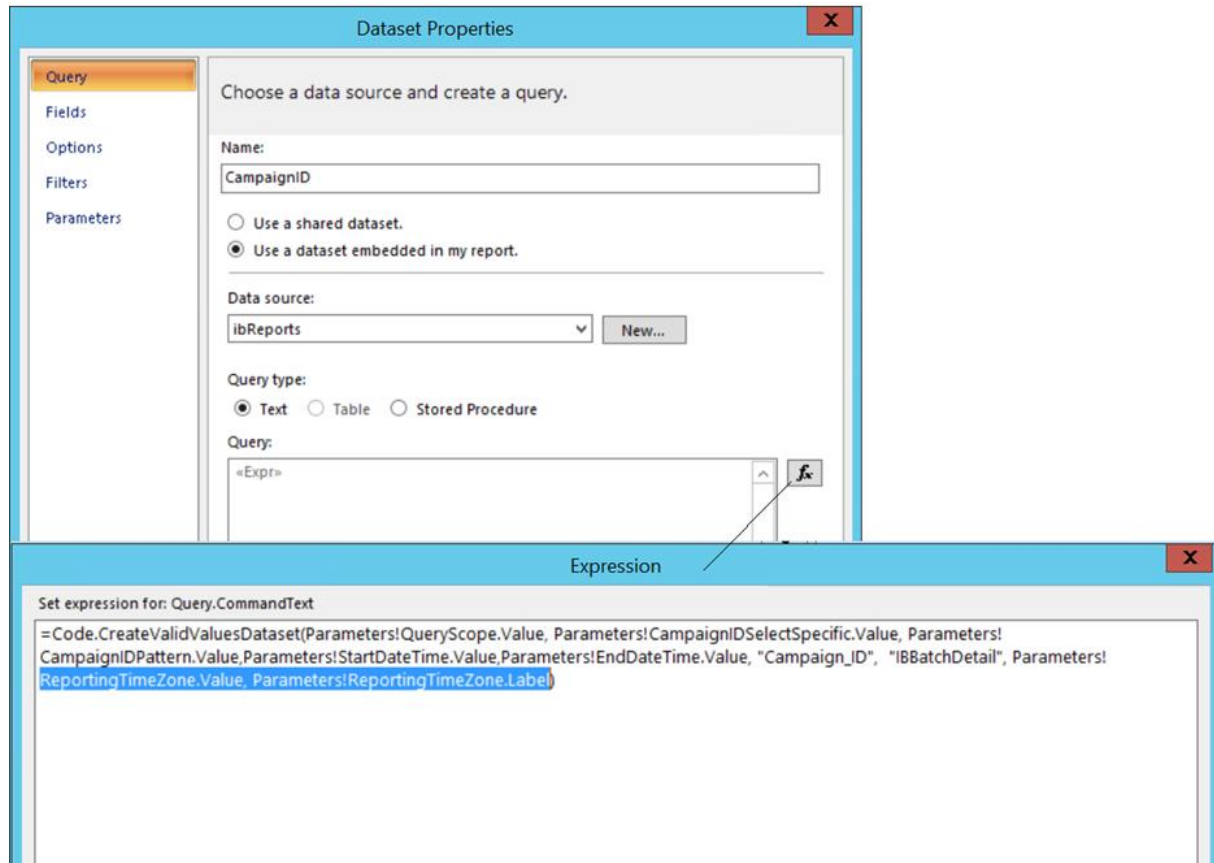
- Select from one of the following options:
 - None
 - Specify values
 - Get values from a query
- Dataset: (Warning: Possible performance impact)
 - ReportingTimeZones
- Value field:
 - tz_id
- Label field:
 - tz_name

At the bottom of the dialog, there are three buttons: "Help", "OK", and "Cancel".

- Use ReportingTimeZones dataset
- Move the ReportingTimeZone parameter up, so that it will list after EndDataTime parameter.

Update all Report Datasets expression

Update all Report Datasets expression call to Code.CreateValidValuesDataset function.



Add new Report Code to Report

Copy and add the new Report Functions code to the custom Report.

New Report Function Code
<pre>Public Function TimeZoneToUtc(ByVal strTimeZoneName As String, _ ByVal strDateTimeLocal As DateTime) As DateTime Dim res As DateTime res = BatchTimeExtensions.BatchTimeFunctions.TimeZoneToUtc(strTimeZoneName, strDateTimeLocal) Return res End function</pre>
<pre>Public Function UtcToTimeZone(ByVal strTimeZoneName As String, _ ByVal strDateTimeUtc As DateTime) As DateTime Dim res As DateTime res = BatchTimeExtensions.BatchTimeFunctions.UtcToTimeZone(strTimeZoneName, strDateTimeUtc) Return res End function</pre>
<pre>Public Function GetTimeZoneDateTime(ByVal strTimeZoneName As String, _ ByVal strDateTimeLocal As DateTime, _ ByVal strDateTimeUtc As DateTime) As String Dim res As String Dim strDateTime = BatchTimeExtensions.BatchTimeFunctions.GetTimeZoneDateTime(strTimeZoneName, strDateTimeLocal, strDateTimeUtc) res = String.Format("{0} {1}", FormatDateTime(strDateTime, 2), FormatDateTime(strDateTime, 3))</pre>

```

Return res
End function

Public Function GetTimeZoneOffset( ByVal strTimeZoneName As String, _
    ByVal strDateTimeLocal As DateTime, _
    ByVal strDateTimeUtc As DateTime ) As String

Dim resOffset As String
resOffset = BatchTimeExtensions.BatchTimeFunctions.GetTimeZoneOffset(strTimeZoneName, strDateTimeLocal, strDateTimeUtc)
Return resOffset
End function

Public Function GetReportDateRange(ByVal strTimeZoneId As String, _
    ByVal strTimeZone As String, _
    ByVal strStartDateTime As DateTime, _
    ByVal strEndDateTime As DateTime ) As String

Dim res As String
res = ""
Dim startDateTime As DateTime
Dim endDateTime As DateTime

If (Trim(strTimeZoneId)="BSLT") Then
    res = String.Format(" (BD.DateTime Between '{0}' AND '{1}')" , Format(strStartDateTime, "s"), Format(strEndDateTime, "s"))
ElseIf (Trim(strTimeZoneId)="UTC") Then
    res = String.Format(" (BD.DateTimeUTC Between '{0}' AND '{1}')" , Format(strStartDateTime, "s"), Format(strEndDateTime, "s"))
Else
    startDateTime = BatchTimeExtensions.BatchTimeFunctions.TimeZoneToUtc(strTimeZone, strStartDateTime)
    endDateTime = BatchTimeExtensions.BatchTimeFunctions.TimeZoneToUtc(strTimeZone, strEndDateTime)
    res = String.Format(" (BD.DateTimeUTC Between '{0}' AND '{1}')" , Format(strStartDateTime, "s"), Format(strEndDateTime,
"s"))
End If
Return res
End function

Public Function GetReportDateColumn( ByVal strTimeZoneId As String, ByVal strColumnPrefix As String) As String

Dim res As String
res = ""
If (Trim(strTimeZoneId) = "BSLT") Then
    res = strColumnPrefix & "DateTime"
Else
    res = strColumnPrefix & "DateTimeUTC"
End If
Return res
End function

Public Function GetTimeZoneSuffix(ByVal strTimeZoneId As String) As String

Dim res As String
res = ""
If (Trim(strTimeZoneId)<>"BSLT") Then
    res = " (" + Trim(strTimeZoneId) + ")"
End If
Return res
End function

Public Function GetServerTimeZoneSuffix() As String

Dim res As String
res = System.Text.RegularExpressions.Regex.Replace(If(System.TimeZone.CurrentTimeZone().IsDaylightSavingTime(now
()),System.TimeZone.CurrentTimeZone().DaylightName,System.TimeZone.CurrentTimeZone().StandardName),"[^Z]", "")
res = " (" + res + ")"
Return res
End function

```

```
Public Function GetFixedStringFormat( ByVal decimalValue As Double ) As String
    Dim resFormatString as String
    resFormatString = BatchTimeExtensions.BatchTimeFunctions.GetFixedStringFormat(decimalValue)
    Return resFormatString
End Function
```

Note: This method is used to display Float values in 8 significant digit format in the Float fields of SSRS reports.

Update Existing Report Code

Copy and replace existing custom report (CreateValidValuesDataset) function with the updated Report Functions code.

Updated Report Function Code

```
Public Function CreateValidValuesDataset(ByVal bQueryScope As Boolean, _
                                        ByVal bSelectSpecific As Boolean, _
                                        ByVal strPattern As String, _
                                        ByVal strStartDateTime As DateTime, _
                                        ByVal strEndDateTime As DateTime, _
                                        ByVal strField As String, _
                                        ByVal strView As String, _
                                        ByVal strTimezoneId As String, _
                                        ByVal strTimezone As String ) As String

    strPattern = ConvertPattern(strPattern)
    Dim strQueryLimit As String
    Dim strReportDateColumn As String
    strQueryLimit = ""
    strReportDateColumn = GetReportDateColumn(strTimezoneId, "")
    Dim res As String
    res = ""
    If (bQueryScope = False) Then
        If (bSelectSpecific = False And strPattern = "") Then
            res = String.Format("SELECT N'ALL' as {0}, 0 as Id ", strField)
        Else
            res = "SELECT "
            If (strQueryLimit <> "") Then
                res = res & String.Format("Top {0} ", strQueryLimit)
            End If

            res = res & String.Format(" (CASE WHEN(Len({0})>0) THEN {1} ELSE N'<Blank>' END) AS {2}, 1 as Id From {3} ", _
                                    strField, _
                                    strField, _
                                    strField, _
                                    strView)

            If (strPattern <> "") Then
                res = res & String.Format("WHERE {0} {1} AND {2} IS NOT NULL AND {3} BETWEEN '{4}'AND '{5}' ", _
                                        strField, _
                                        strPattern, _
                                        strField, _
                                        strReportDateColumn, _
                                        Format(UtcToTimeZone(strTimezone, strStartDateTime), "s"), _
```

```

Format(UtcToTimeZone(strTimeZone, strEndDateTime), "s"))

Else
    res = res & String.Format(" WHERE {0} IS NOT NULL AND {1} BETWEEN '{2}' AND '{3}' ", strField , strReportDateColumn, _
    Format(UtcToTimeZone(strTimeZone, strStartDateTime), "s"), Format(UtcToTimeZone(strTimeZone, strEndDateTime), "s"))
End If

res = res & String.Format(" UNION Select N'ALL' AS {0}, 0 as Id order by Id, {1}", strField, strField)
End If
Else
res = String.Format("SELECT N'ALL' as {0}, 0 as Id ", strField)
End If

CreateValidValuesDataset = res
End Function
    
```

Other Report updates


Item for Update (if applicable)	Expression
Update WhereClause Parameter to use new report GetReportDateRange code	<pre> =Code.WhereCondition(Parameters!CampaignIDPattern.Value, "Campaign_ID", Parameters!CampaignID.Value) & Code.WhereCondition(Parameters!LotIDPattern.Value, "Lot_ID", Parameters!LotID.Value) & Code.WhereCondition(Parameters!BatchIDPattern.Value, "Batch_ID", Parameters!BatchID.Value) & Code.WhereCondition(Parameters!BatchServerNamePattern.Value, "Batch_Server_Name", Parameters!BatchServerName.Value) & Code.WhereCondition(Parameters!ConnectionNamePattern.Value, "UnitOrConnection", Parameters!ConnectionName.Value) & Code.WhereCondition(Parameters!ProcedureIDPattern.Value, "UnitProcedure_ID", Parameters!ProcedureID.Value) & Code.WhereCondition(Parameters!OperationIDPattern.Value, "Operation_ID", Parameters!OperationID.Value) & Code.WhereCondition(Parameters!PhaseIDPattern.Value, "Phase_ID", Parameters!PhaseID.Value) & Code.WhereCondition(Parameters!PhaseLabelPattern.Value, "DoneBy_User_ID", Parameters!PhaseLabel.Value) & Code.WhereCondition(Parameters!DoneByUserIDPattern.Value, "Phase_Label", Parameters!DoneByUserID.Value) & Code.WhereCondition(Parameters!CheckedByUserIDPattern.Value, "CheckBy_User_ID", Parameters!CheckedByUserID.Value) & Code.WhereCondition(Parameters!DescriptionPattern.Value, "Description", Parameters!Description.Value) & Code.GetReportDateRange(Parameters!ReportingTimeZone.Value, Parameters!ReportingTimeZone.Label, Parameters!StartDateTime.Value, Parameters!EndDateTime.Value) </pre>
Update Time Period display to include Time zone suffix	<pre> =FORMAT(First(Fields!StartDateTimeLocal.Value, "TimeZone"), "G") & " - " & FORMAT(First(Fields!EndDateTimeLocal.Value, "TimeZone"), "G") Change to: =Parameters!StartDateTime.Value & " - " & Parameters!EndDateTime.Value & Code.GetTimeZoneSuffix(Parameters!ReportingTimeZone.Value) </pre>
Any Detail Report Heading that display date time. Example IBBatchReport DateTime	<pre> ="Date/Time" & Code.GetTimeZoneSuffix(Parameters!ReportingTimeZone.Value) </pre>

For Report Detail column that display date time.	=Code.GetTimeZoneDateTime(Parameters!ReportingTimeZone.Label, Fields!DateTime.Value, Fields!DateTimeUTC.Value) & " " & Code.GetTimeZoneOffset(Parameters!ReportingTimeZone.Label, Fields!DateTime.Value, Fields!DateTimeUTC.Value)
Remove Deprecated WIS Report Items	<p>Deprecated Datasets</p> <ul style="list-style-type: none"> LayoutDurationChoices, LayoutTimeZoneInfo, TimeZone (If applicable) <p>Deprecated Parameters</p> <ul style="list-style-type: none"> Duration, TimeZoned (If applicable)
Update references to LayoutTimeZoneInfo: StartDateTimeLocal.Value	<p>=FORMAT(First(Fields!StartDateTimeLocal.Value, "LayoutTimeZoneInfo"), "yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffk")</p> <p>Change to:</p> <p>=FORMAT(Parameters! StartDateTime.Value, "yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffk")</p>
Update references to LayoutTimeZoneInfo: EndDateTimeUTC.Value	<p>=FORMAT(First(Fields!EndDateTimeUTC.Value, "LayoutTimeZoneInfo"), "yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffk")</p> <p>Change to:</p> <p>=FORMAT(Parameters!EndDateTime.Value, "yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffk")</p>
	<p>=FORMAT(First(Fields!StartDateTimeUTC.Value, "LayoutTimeZoneInfo"), "yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffk")</p> <p>Change to:</p> <p>=FORMAT(Parameters! StartDateTime.Value, "yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffk")</p>
	<p>=FORMAT(First(Fields!EndDateTimeUTC.Value, "LayoutTimeZoneInfo"), "yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffk")</p> <p>Change to:</p> <p>=FORMAT(Parameters!EndDateTime.Value, "yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffk")</p>
Remove unused datasets	Remove unused datasets that use aaReport Data source
Remove unused report data source	Delete aaReports from Report
Add Time zone suffix to "Generated On" Date time	<p>Change:</p> <p>= "Report Generated On " + FormatDateTime(Globals!ExecutionTime)</p> <p>To:</p> <p>= "Report Generated On " + FormatDateTime(Globals!ExecutionTime) + Code.GetServerTimeZoneSuffix()</p>

Reporting in Time Zone

Start DateTime:	<input type="text" value="2/10/2016 1:11:49 PM"/>	End DateTime:	<input type="text" value="2/11/2016 1:11:49 PM"/>
Reporting Time Zone:	<ul style="list-style-type: none">Batch Server Local TimeUniversal TimeEastern Standard TimePacific Standard TimeEastern Daylight TimeCentral European TimeEastern European Time	Do not auto-populate list boxes:	<input checked="" type="radio"/> True <input type="radio"/> False
BatchServerName Pattern:	<input type="text"/>	CampaignID Pattern:	<input type="text"/>
Select Specific BatchServerName:	<input type="text"/>	Select Specific CampaignID:	<input checked="" type="radio"/> True <input type="radio"/> False
BatchServerName List:	<input type="text" value="ALL"/>	CampaignID List:	<input type="text" value="ALL"/>
LotID Pattern:	<input type="text"/>	BatchID Pattern:	<input type="text"/>
Select Specific LotID:	<input checked="" type="radio"/> True <input type="radio"/> False	Select Specific BatchID:	<input checked="" type="radio"/> True <input type="radio"/> False
LotID List:	<input type="text" value="ALL"/>	BatchID List:	<input type="text" value="ALL"/>
UnitOrConnection Pattern:	<input type="text"/>	ProcedureID Pattern:	<input type="text"/>

Sample Report with Time Zone Information



Batch Detail
 Batch Management Reports

Filter Criteria

Time Period: 7/15/2019 2:05:32 PM - 7/15/2019 2:09:32 PM

Batch Server Name:

CampaignID:

LotID:

BatchID:

UnitOrConnection:

ProcedureID:

OperationID:

PhaseID:

Phase Label:

Done By UserID:

Check By UserID:

Description:

Outlining: Expanded

☐ C7/L/B

Date/Time	Equipment	Unit Procedure	Operation	Phase	Description
7/15/2019 2:08:34 PM (UTC+5.5)					Batch received Start
7/15/2019 2:08:34 PM (UTC+5.5)					Batch set Run
7/15/2019 2:08:34 PM (UTC+5.5)		1			Unit Procedure Received Run
7/15/2019 2:08:34 PM (UTC+5.5)		1	BulkTransfer		Operation Received Run
7/15/2019 2:08:34 PM (UTC+5.5)	R210				Allocate
7/15/2019 2:08:34 PM (UTC+5.5)	T110R210				Allocate
7/15/2019 2:08:34 PM (UTC+5.5)	T120R210				Allocate
7/15/2019 2:08:34 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Received Ready
7/15/2019 2:08:34 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Received Ready
7/15/2019 2:08:34 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Set Start
7/15/2019 2:08:34 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Set Start
7/15/2019 2:08:34 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Received Run
7/15/2019 2:08:34 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Received Run
7/15/2019 2:08:34 PM (UTC+5.5)					Batch received Start
7/15/2019 2:08:34 PM (UTC+5.5)					Batch set Run
7/15/2019 2:08:34 PM (UTC+5.5)		1			Unit Procedure Received Run
7/15/2019 2:08:34 PM (UTC+5.5)		1	BulkTransfer		Operation Received Run
7/15/2019 2:08:34 PM (UTC+5.5)	R210				Allocate
7/15/2019 2:08:34 PM (UTC+5.5)	T110R210				Allocate
7/15/2019 2:08:34 PM (UTC+5.5)	T120R210				Allocate
7/15/2019 2:08:34 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Received Ready
7/15/2019 2:08:34 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Received Ready
7/15/2019 2:08:34 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Set Start
7/15/2019 2:08:34 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Set Start
7/15/2019 2:08:34 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Received Run
7/15/2019 2:08:34 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Received Run
7/15/2019 2:08:35 PM (UTC+5.5)					Batch received Hold
7/15/2019 2:08:35 PM (UTC+5.5)					Batch set Held
7/15/2019 2:08:35 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Set Hold
7/15/2019 2:08:35 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Set Hold
7/15/2019 2:08:36 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Received Held
7/15/2019 2:08:36 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Received Held
7/15/2019 2:08:38 PM (UTC+5.5)					Batch received Abort
7/15/2019 2:08:38 PM (UTC+5.5)					Batch set Aborting
7/15/2019 2:08:38 PM (UTC+5.5)		1			Unit Procedure Received Done
7/15/2019 2:08:38 PM (UTC+5.5)		1	BulkTransfer		Operation Received Done
7/15/2019 2:08:38 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Set Abort
7/15/2019 2:08:38 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Set Abort
7/15/2019 2:08:38 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Received Aborted
7/15/2019 2:08:38 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Received Aborted
7/15/2019 2:08:38 PM (UTC+5.5)	T110R210	1	BulkTransfer	INGR_ADD	Set Reset
7/15/2019 2:08:38 PM (UTC+5.5)	T110R210				Release
7/15/2019 2:08:38 PM (UTC+5.5)	T120R210	1	BulkTransfer	INGR_ADD	Set Reset
7/15/2019 2:08:38 PM (UTC+5.5)	T120R210				Release
7/15/2019 2:08:38 PM (UTC+5.5)	R210				Release
7/15/2019 2:08:38 PM (UTC+5.5)					Batch set Aborted

Report Generated On 7/15/2019 2:10:02 PM (IST)

Page 1 / 1

