Manitou Core - SQL Query - Go Live Checklist

12/27/2023 5:57 pm EST

Go Live Checklist

Pre Go Live Checklist

- Are all receivers communicating with Manitou?
- Is Replication running?
- Is all of the site's system information recorded in the Bold Support Portal (Fusion\Kayako)
- Are all Manitou Service able to run?
- Download the GoLive scripts to the customer site
- Is there any cleanup in the MSM needed?
- Has the customer been upgraded to the latest patch version? If not, upgrade them.

Go Live Procedures

- Connect to the Primary Manitou Server
- Stop all Manitou Services
- Delete any .sig files found in the FEP Files folder
- Run the Truncate Tables Go Live Script
- Open Microsoft SQL Server Management Studio and Login

Open the "Go Live - Truncate Tables" in a query window

```
-------------------------------------------------------------------------
--    Go Live - Truncate Tables script
-- Truncate tables before starting the system for "Go Live"
--
-- Step 1: Stop the Broker and all FEPs
-- Step 2: Run this script on the active database server.
--         You may do this while replication is running.
-- Step 3: IMPORTANT: Don't forget to delete the .sig files from the FEPs before you restart the system or else a batch of ol
d signals may be delivered.
-- Step 4: Restart Manitou and handle signals. Hooray! You're LIVE!
--
-- This script deletes all log tables (CLOG, BLOG, SYSLOG, etc),
-- alarm tables, and status tables.
--
-------------------------------------------------------------------------
USE Manitou
go

declare @tablename varchar(1000), @sql varchar(1000)

declare tables cursor for
select table_name from information_schema.tables t
where t.table_name like '[BCMS]LOG%'
or t.table_name like 'SYSLOG%'
or t.table_name like 'RAWDATA%'
or t.table_name like 'AUDIT%'
or t.table_name like 'ALARM%'
or t.table_name in ('ACTIONSEXEC','OUTSRV','PENDING','PRECANCEL','SIGTRANS','STXSTS')
open tables

fetch next from tables into @tablename
while @@FETCH_STATUS=0
begin
      set @sql = 'DELETE FROM ' + @tablename
      print 'EXEC ' + @sql
      exec(@sql)
      fetch next from tables into @tablename
end

close tables
deallocate tables

DELETE FROM COMMENTS WHERE COMMENTTYPE=1000
```

- Change USE SOMEDATABASE at the top of the script to USE MANITOU
- Run the script
- Next Run the "Transmitter Test Go Live" Script

```
USE somedatabase

SET noexec off
go

--Make sure you update the @SeedDtTm on line 523
IF OBJECT_ID(N'TZOffset', N'FN') > 0
 DROP FUNCTION dbo.TZOffset
```

```
go

/****** Object:  UserDefinedFunction [dbo].[TZOffset]    Script Date: 5/15/2019 10:57:18 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION [dbo].[TZOffset]
 (@dt datetime, @SrcTimezone varchar(15), @OutTimezone varchar(15))
RETURNS datetime
AS
BEGIN
 declare @OFFSET smallint, @DSBEGIN smallint, @DSBEGINDY smallint, @DSBEGINDOM smallint,
  @DSBEGINMT smallint, @DSBEGINTM datetime, @DSEND smallint, @DSENDDY smallint,
  @DSENDDOM smallint, @DSENDMT smallint, @DSENDTM datetime, @DSMINS smallint, @DAYSAVE smallint,
  @Year int, @WorkDt varchar(10), @DateFmt varchar(3),
  @DSBeginDt datetime, @DSEndDt datetime

 set @DateFmt = 'mdy'

-- Convert source event date/time to UTC date/time
 if @SrcTimezone = 'UTC' -- already have GMT0 date/time
 begin
  set @OFFSET = 0
  goto SkipDS
 end

 select @DAYSAVE = daysave, @OFFSET = offset, @DSBEGIN = dsbegin, @DSBEGINDY = dsbegindy,
  @DSBEGINDOM = dsbegindom, @DSBEGINMT = dsbeginmt, @DSBEGINTM = dsbegintm, @DSEND = dsend,
  @DSENDDY = dsenddy, @DSENDDOM = dsenddom, @DSENDMT = dsendmt, @DSENDTM = dsendtm,
  @DSMINS = dsmins
 from MANITOU..TIMEZONE with (readuncommitted)
 where timezone = @SrcTimezone and (validfrm is null or @dt >= validfrm) and
  (validto is null or @dt <= validto)

 if @OFFSET is null
  set @OFFSET = 0

 set @Year = year(@dt)
 if @Year < 1970  -- Manitou doesn't calculate DS for years prior to 1970.
  goto SkipDS

 if @DSMINS = 0  -- Source timezone does not have DS offset
  goto SkipDS

 -- Check if event date is in Daylight Saving.
 if @DAYSAVE <> 1  -- Source timezone does not participate in Daylight Saving
  goto SkipDS

 -- Calculate DS begin date in the event year.  'Begin' options as follows:
 -- 0 = Beginning - e.g. first Sunday in April
 -- 2 = Specific date - e.g. March 12
 if @DSBEGIN = 0 or @DSBEGIN = 2
 begin
  -- Construct initial value of DS begin date
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then @DSBEGINDOM
    when 'm' then @DSBEGINMT
```

```
   else @Year
  end
 set @WorkDt = @WorkDt + '/'
 set @WorkDt =
  case substring(@DateFmt, 2, 1)
   when 'd' then @WorkDt + cast(@DSBEGINDOM as varchar(2))
   when 'm' then @WorkDt + cast(@DSBEGINMT as varchar(2))
   else @WorkDt + cast(@Year as varchar(4))
  end
 set @WorkDt = @WorkDt + '/'
 set @WorkDt =
  case right(@DateFmt, 1)
   when 'd' then @WorkDt + cast(@DSBEGINDOM as varchar(2))
   when 'm' then @WorkDt + cast(@DSBEGINMT as varchar(2))
   else @WorkDt + cast(@Year as varchar(4))
  end
 set @DSBeginDt = @WorkDt

 if @DSBEGIN = 0
  -- Increment date until we are at appropriate day of week.
  while datepart(dw, @DSBeginDt) <> @DSBEGINDY
   set @DSBeginDt = dateadd(dd, 1, @DSBeginDt)
end
else  -- Daylight Saving begins on the last occurrence of a given day of the month.
begin
 -- Construct initial value of DS begin date.
 -- Begin at end of month (actually first day of next month).
 if @DSBEGINMT < 12
 begin
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then '1'
    when 'm' then (@DSBEGINMT + 1)
    else @Year
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case substring(@DateFmt, 2, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + cast(@DSBEGINMT + 1 as varchar(2))
    else @WorkDt + cast(@Year as varchar(4))
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case right(@DateFmt, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + cast(@DSBEGINMT + 1 as varchar(2))
    else @WorkDt + cast(@Year as varchar(4))
   end
 end
 else
 begin
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then '1'
    when 'm' then '1'
    else (@Year + 1)
   end
  set @WorkDt = @WorkDt + '/'
```

```
  set @WorkDt =
   case substring(@DateFmt, 2, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + '1'
    else @WorkDt + cast(@Year + 1 as varchar(4))
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case right(@DateFmt, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + '1'
    else @WorkDt + cast(@Year + 1 as varchar(4))
   end
 end
 set @DSBeginDt = @WorkDt

 -- We now have first day of month after DS begins.
 -- Set to last day of actual DS begin month.
 set @DSBeginDt = dateadd(dd, -1, @DSBeginDt)

 -- Decrement date until we are at appropriate day of week.
 while datepart(dw, @DSBeginDt) <> @DSBEGINDY
  set @DSBeginDt = dateadd(dd, -1, @DSBeginDt)
end
-- We have the date, now set time of day.
set @DSBeginDt = dateadd(hh, datepart(hh, @DSBEGINTM), @DSBeginDt)
set @DSBeginDt = dateadd(mi, datepart(mi, @DSBEGINTM), @DSBeginDt)

-- Calculate DS end date in the event year.  'End' options as follows:
-- 0 = Beginning - e.g. first Sunday in October
-- 2 = Specific date - e.g. October 12
if @DSEND = 0 or @DSEND = 2
begin
 -- Construct initial value of DS end date
 set @WorkDt =
  case left(@DateFmt, 1)
   when 'd' then @DSENDDOM
   when 'm' then @DSENDMT
   else @Year
  end
 set @WorkDt = @WorkDt + '/'
 set @WorkDt =
  case substring(@DateFmt, 2, 1)
   when 'd' then @WorkDt + cast(@DSENDDOM as varchar(2))
   when 'm' then @WorkDt + cast(@DSENDMT as varchar(2))
   else @WorkDt + cast(@Year as varchar(4))
  end
 set @WorkDt = @WorkDt + '/'
 set @WorkDt =
  case right(@DateFmt, 1)
   when 'd' then @WorkDt + cast(@DSENDDOM as varchar(2))
   when 'm' then @WorkDt + cast(@DSENDMT as varchar(2))
   else @WorkDt + cast(@Year as varchar(4))
  end
 set @DSEndDt = @WorkDt

 if @DSEND = 0
  -- Increment date until we are at appropriate day of week.
  while datepart(dw, @DSEndDt) <> @DSENDDY
   set @DSEndDt = dateadd(dd, 1, @DSEndDt)
```

```
end
else  -- Daylight Saving ends on the last occurrence of a given day of the month.
begin
 -- Construct initial value of DS end date.
 -- Begin at end of month (actually first day of next month).
 if @DSENDMT < 12
 begin
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then '1'
    when 'm' then (@DSENDMT + 1)
    else @Year
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case substring(@DateFmt, 2, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + cast(@DSENDMT + 1 as varchar(2))
    else @WorkDt + cast(@Year as varchar(4))
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case right(@DateFmt, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + cast(@DSENDMT + 1 as varchar(2))
    else @WorkDt + cast(@Year as varchar(4))
   end
 end
 else
 begin
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then '1'
    when 'm' then '1'
    else (@Year + 1)
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case substring(@DateFmt, 2, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + '1'
    else @WorkDt + cast(@Year + 1 as varchar(4))
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case right(@DateFmt, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + '1'
    else @WorkDt + cast(@Year + 1 as varchar(4))
   end
 end
 set @DSEndDt = @WorkDt

 -- We now have first day of month after DS ends.
 -- Set to last day of actual DS end month.
 set @DSEndDt = dateadd(dd, -1, @DSEndDt)

 -- Decrement date until we are at appropriate day of week.
 while datepart(dw, @DSEndDt) <> @DSENDDY
```

```
  set @DSEndDt = dateadd(dd, -1, @DSEndDt)
 end
 -- We have the date, now set time of day.
 set @DSEndDt = dateadd(hh, datepart(hh, @DSENDTM), @DSEndDt)
 set @DSEndDt = dateadd(mi, datepart(mi, @DSENDTM), @DSEndDt)

 -- If the event date falls within DS, add the DS offset.
 if @DSBeginDt < @DSEndDt  -- northern hemisphere
 begin
  if dateadd(mi, -(@DSMINS), @dt) >= @DSBeginDt and @dt < @DSEndDt
  begin
   set @Offset = @Offset + @DSMINS
  end
 end
 else  -- southern hemisphere
 begin
  if dateadd(mi, -(@DSMINS), @dt) >= @DSBeginDt or @dt < @DSEndDt
  begin
   set @Offset = @Offset + @DSMINS
  end
 end
SkipDS:
 if @Offset <> 0
  set @dt = dateadd(mi, -(@Offset), @dt)

-- now convert UTC date/time to destination timezone
 if @OutTimezone = 'UTC' -- already have it
  goto SkipDS2

 select @DAYSAVE = daysave, @OFFSET = offset, @DSBEGIN = dsbegin, @DSBEGINDY = dsbegindy,
  @DSBEGINDOM = dsbegindom, @DSBEGINMT = dsbeginmt, @DSBEGINTM = dsbegintm, @DSEND = dsend,
  @DSENDDY = dsenddy, @DSENDDOM = dsenddom, @DSENDMT = dsendmt, @DSENDTM = dsendtm,
  @DSMINS = dsmins
 from Manitou..TIMEZONE with (readuncommitted)
 where timezone = @OutTimezone and (validfrm is null or @dt >= validfrm) and
  (validto is null or @dt <= validto)

 if @OFFSET is null
  goto SkipDS2

 if @Offset <> 0
  set @dt = dateadd(mi, @Offset, @dt)

 set @Year = year(@dt)
 if @Year < 1970  -- Manitou doesn't calculate DS for years prior to 1970.
  goto SkipDS2

 if @DSMINS = 0  -- Destination timezone does not have DS offset
  goto SkipDS2

 -- Check if date is in Daylight Saving.
 if @DAYSAVE <> 1  -- Destination timezone does not participate in Daylight Saving
  goto SkipDS2

 -- Calculate DS begin date in the event year.  'Begin' options as follows:
 -- 0 = Beginning - e.g. first Sunday in April
 -- 2 = Specific date - e.g. March 12
 if @DSBEGIN = 0 or @DSBEGIN = 2
 begin
  -- Construct initial value of DS begin date
```

```
 set @WorkDt =
  case left(@DateFmt, 1)
   when 'd' then @DSBEGINDOM
   when 'm' then @DSBEGINMT
   else @Year
  end
 set @WorkDt = @WorkDt + '/'
 set @WorkDt =
  case substring(@DateFmt, 2, 1)
   when 'd' then @WorkDt + cast(@DSBEGINDOM as varchar(2))
   when 'm' then @WorkDt + cast(@DSBEGINMT as varchar(2))
   else @WorkDt + cast(@Year as varchar(4))
  end
 set @WorkDt = @WorkDt + '/'
 set @WorkDt =
  case right(@DateFmt, 1)
   when 'd' then @WorkDt + cast(@DSBEGINDOM as varchar(2))
   when 'm' then @WorkDt + cast(@DSBEGINMT as varchar(2))
   else @WorkDt + cast(@Year as varchar(4))
  end
 set @DSBeginDt = @WorkDt

 if @DSBEGIN = 0
  -- Increment date until we are at appropriate day of week.
  while datepart(dw, @DSBeginDt) <> @DSBEGINDY
   set @DSBeginDt = dateadd(dd, 1, @DSBeginDt)
end
else  -- Daylight Saving begins on the last occurrence of a given day of the month.
begin
 -- Construct initial value of DS begin date.
 -- Begin at end of month (actually first day of next month).
 if @DSBEGINMT < 12
 begin
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then '1'
    when 'm' then (@DSBEGINMT + 1)
    else @Year
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case substring(@DateFmt, 2, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + cast(@DSBEGINMT + 1 as varchar(2))
    else @WorkDt + cast(@Year as varchar(4))
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case right(@DateFmt, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + cast(@DSBEGINMT + 1 as varchar(2))
    else @WorkDt + cast(@Year as varchar(4))
   end
 end
 else
 begin
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then '1'
```

```
    when 'm' then '1'
    else (@Year + 1)
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case substring(@DateFmt, 2, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + '1'
    else @WorkDt + cast(@Year + 1 as varchar(4))
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case right(@DateFmt, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + '1'
    else @WorkDt + cast(@Year + 1 as varchar(4))
   end
 end
 set @DSBeginDt = @WorkDt

 -- We now have first day of month after DS begins.
 -- Set to last day of actual DS begin month.
 set @DSBeginDt = dateadd(dd, -1, @DSBeginDt)

 -- Decrement date until we are at appropriate day of week.
 while datepart(dw, @DSBeginDt) <> @DSBEGINDY
  set @DSBeginDt = dateadd(dd, -1, @DSBeginDt)
end
-- We have the date, now set time of day.
set @DSBeginDt = dateadd(hh, datepart(hh, @DSBEGINTM), @DSBeginDt)
set @DSBeginDt = dateadd(mi, datepart(mi, @DSBEGINTM), @DSBeginDt)

-- Calculate DS end date in the event year.  'End' options as follows:
-- 0 = Beginning - e.g. first Sunday in October
-- 2 = Specific date - e.g. October 12
if @DSEND = 0 or @DSEND = 2
begin
 -- Construct initial value of DS end date
 set @WorkDt =
  case left(@DateFmt, 1)
   when 'd' then @DSENDDOM
   when 'm' then @DSENDMT
   else @Year
  end
 set @WorkDt = @WorkDt + '/'
 set @WorkDt =
  case substring(@DateFmt, 2, 1)
   when 'd' then @WorkDt + cast(@DSENDDOM as varchar(2))
   when 'm' then @WorkDt + cast(@DSENDMT as varchar(2))
   else @WorkDt + cast(@Year as varchar(4))
  end
 set @WorkDt = @WorkDt + '/'
 set @WorkDt =
  case right(@DateFmt, 1)
   when 'd' then @WorkDt + cast(@DSENDDOM as varchar(2))
   when 'm' then @WorkDt + cast(@DSENDMT as varchar(2))
   else @WorkDt + cast(@Year as varchar(4))
  end
 set @DSEndDt = @WorkDt
```

```
 if @DSEND = 0
  -- Increment date until we are at appropriate day of week.
  while datepart(dw, @DSEndDt) <> @DSENDDY
   set @DSEndDt = dateadd(dd, 1, @DSEndDt)
end
else  -- Daylight Saving ends on the last occurrence of a given day of the month.
begin
 -- Construct initial value of DS end date.
 -- Begin at end of month (actually first day of next month).
 if @DSENDMT < 12
 begin
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then '1'
    when 'm' then (@DSENDMT + 1)
    else @Year
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case substring(@DateFmt, 2, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + cast(@DSENDMT + 1 as varchar(2))
    else @WorkDt + cast(@Year as varchar(4))
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case right(@DateFmt, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + cast(@DSENDMT + 1 as varchar(2))
    else @WorkDt + cast(@Year as varchar(4))
   end
 end
 else
 begin
  set @WorkDt =
   case left(@DateFmt, 1)
    when 'd' then '1'
    when 'm' then '1'
    else (@Year + 1)
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case substring(@DateFmt, 2, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + '1'
    else @WorkDt + cast(@Year + 1 as varchar(4))
   end
  set @WorkDt = @WorkDt + '/'
  set @WorkDt =
   case right(@DateFmt, 1)
    when 'd' then @WorkDt + '1'
    when 'm' then @WorkDt + '1'
    else @WorkDt + cast(@Year + 1 as varchar(4))
   end
 end
 set @DSEndDt = @WorkDt

 -- We now have first day of month after DS ends.
 -- Set to last day of actual DS end month.
```

```
 set @DSEndDt = dateadd(dd, -1, @DSEndDt)

 -- Decrement date until we are at appropriate day of week.
 while datepart(dw, @DSEndDt) <> @DSENDDY
  set @DSEndDt = dateadd(dd, -1, @DSEndDt)
end
-- We have the date, now set time of day.
set @DSEndDt = dateadd(hh, datepart(hh, @DSENDTM), @DSEndDt)
set @DSEndDt = dateadd(mi, datepart(mi, @DSENDTM), @DSEndDt)

-- If the date falls within DS, add the DS offset.
if @DSBeginDt < @DSEndDt  -- northern hemisphere
begin
 if dateadd(mi, @DSMINS, @dt) >= @DSBeginDt and @dt < @DSEndDt
 begin
  set @dt = dateadd(mi, @DSMINS, @dt)
 end
end
else  -- southern hemisphere
begin
 if dateadd(mi, @DSMINS, @dt) >= @DSBeginDt or @dt < @DSEndDt
 begin
  set @dt = dateadd(mi, @DSMINS, @dt)
 end
end
SkipDS2:
 return @dt
END
go

--!!!! Make sure you set the variables below before you execute this!
declare @SeedDtTm datetime

--This is the time when the late to tests will come in.
--Set the day portion to the day of the go live
--Set the time to whatever time of day they want the late to test signals to come in
--    This will convert the time you input into gmt for the table
set @SeedDtTm = '02/07/2019 21:00'

if @SeedDtTm < GETDATE()
 BEGIN
  SELECT 'Make sure the seed date time is not in the past'
  set noexec on
 END

select @SeedDtTm = dbo.TZOffset(@SeedDtTm, t.TIMEZONE, 'UTC')
 from contact t
 where t.CONTTYPE = 0

update stx set nexttest =
case
    when testunit = 0 then dateadd(mi, testintvl, @SeedDtTm)
    when testunit = 1 then dateadd(hh, testintvl, @SeedDtTm)
    else dateadd(dd, testintvl, @SeedDtTm)
end
where testintvl is not null and testintvl > 0 and (nexttest is null or nexttest <
    case
        when testunit = 0 then dateadd(mi, testintvl, @SeedDtTm)
        when testunit = 1 then dateadd(hh, testintvl, @SeedDtTm)
        else dateadd(dd, testintvl, @SeedDtTm)
```

```
    end)
```

- Open the "Go Live - Transmitter Test.txt" in a query window
- Change USE SOMEDATABASE at the top of the script to USE MANITOU
- Change the @SeedDtTm = '02/07/2019 21:00' on line 522 to the day and time when they handle their late to tests.
    - If you don't have line numbers once you paste the script, scroll all the way to the end and then scroll up until you find the line.
    - If they are flipping Y cables on the day they go live, then set the date to that day.
    - Iif they are waiting to take the y cables or port splitters out of the equation then set it to the day that they're doing that.
- Run the script to make changes to the transmitter tests
- Now Start the Manitou Services
- Verify the customer is receiving signals, and that the operators can log in.
- Update the license to expire 185 days from the Go Live date
- Send the exact time and time zone of the first signal received to the project manager.
- If the customer is ready, assist with flipping the y-cables
- Verify the customer has no other questions, if not work on transitioning them to support.