Update Database_EN_2016

Last Update :	14 Oct. 16
Revision :	1.0a
Content :	Describe how to use the scripts to update database columns

1 Content

2 Summary

How to use the scripts to migrate database from old PcVue version to PcVue 11.2

3 Detail

There are Sql scripts provided in « bin » folder of PcVue that convert old SQL Database format to the new one. The scripts are:

- UpgradeLogTable.sql
- UpgradeTrendTable.sql

4 How to use the two scripts

4.1 Important remark

If you have several tables, you will have to execute these scripts for each of them. But in this case, you have to **remove the Step 5**.
This step will update the "AICommonParameters" table so in the next execution the script will not be executed.

After having updated every tables, you have to execute the step 5 of the query that will update the version in the table AICommonParameters

4.2 Upgrade the Log Table

```
-- UpgradeLogTable.sql

USE [<DatabaseName, sysname,>]

/*================================================================================
 |                       BEFORE EXECUTING THIS SCRIPT                             |
 |                                                                                |
 | Use this template file to manually update an HDS LOG table.                    |
 | Do not use this template file to update an HDS TREND table.                    |
 |                                                                                |
 | 1) Open this file with Sql Server management Studio (SSMS)                     |
 | 2) Use 'specify values for template parameters ' as following                  |
 |    Menu : Query -> Specify values for template parameters (Ctrl+Shift+M)       |
 | 3) Enter DatabaseName and TableName to update the script                       |
 | 4) Click on Execute as following :                                             |
 |    Menu : Query -> Execute (F5)                                                |
 | 5) Wait 'Update finished' message                                              |
  ================================================================================*/


/* ********************************************** */
/* --- STEP 1 : RETRIEVE DB VERSION NUMBER    --- */
/* ********************************************** */
PRINT 'Retreive datebase version'
DECLARE @CurrentDBVersion INT;
DECLARE @DBVersionToReach INT;
DECLARE @NeedsIndexRebuild BIT;

SET @CurrentDBVersion = (SELECT TOP 1 DbVersion FROM AICommonParameters)
SET @DBVersionToReach = 3

PRINT 'Current version : ' + CONVERT(varchar(1), @CurrentDBVersion)

/* --- remove this part bellow if one of your tables has already been set to the current
version --- */
IF @CurrentDBVersion >= @DBVersionToReach
BEGIN
PRINT 'Current table version is up to date'
PRINT 'No further treatment required'
PRINT 'Update finished'
RETURN
END
/* --- remove this part above if one of your tables has already been set to the current
version --- */

IF @CurrentDBVersion = 1
BEGIN
      SET @NeedsIndexRebuild = 1;
      PRINT 'Version compatibility needs to rebuild the index to properly process'
      PRINT 'This task may take a while'
END

/* ****************************************************** */
/* --- STEP 2 : DROP INDEX ON TABLE TO UPDATE IF REQUIRED --- */
/* ****************************************************** */

IF @NeedsIndexRebuild = 1
BEGIN
```

```
      PRINT 'Drop index on table to update'
      ALTER TABLE <TableName,sysname,>
      DROP CONSTRAINT IX_LOG_<TableName,sysname,>_PRIMARY
END

/* ******************************************** */
/* -----    STEP 3 : UPDATE COLUMNS SIZE    ----- */
/* ******************************************** */
BEGIN
      PRINT 'Begin update on columns size'
      IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'Name')
      BEGIN
            PRINT 'Update <Name> column size to varchar 255'
            ALTER TABLE [<TableName,sysname,>]
            ALTER COLUMN [Name] varchar(255) NOT NULL
      END

      IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'Nature')
      BEGIN
            PRINT 'Update <Nature> column size to varchar 100'
            ALTER TABLE [<TableName,sysname,>]
            ALTER COLUMN [Nature] varchar(100) NULL
      END

      IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'Domain')
      BEGIN
            PRINT 'Update <Domain> column size to varchar 100'
            ALTER TABLE [<TableName,sysname,>]
            ALTER COLUMN [Domain] varchar(100) NULL
      END

      IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'Description')
      BEGIN
            PRINT 'Update <Description> column size to varchar 255'
            ALTER TABLE [<TableName,sysname,>]
            ALTER COLUMN [Description] varchar(255) NULL
      END

      IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'AssocLabel')
      BEGIN
            PRINT 'Update <AssocLabel> column size to varchar 255'
            ALTER TABLE [<TableName,sysname,>]
            ALTER COLUMN [AssocLabel] varchar(255) NULL
      END

      IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'EvtTitle')
      BEGIN
            PRINT 'Update <EvtTitle> column size to varchar 255'
            ALTER TABLE [<TableName,sysname,>]
            ALTER COLUMN [EvtTitle] varchar(255) NOT NULL
      END

      DECLARE @sqlRequest NVARCHAR(max)
      DECLARE @counter INT;
      DECLARE @textAttrName varchar(10)

      SET @counter = 3;
```

```
	WHILE (@counter < 17)
	BEGIN
		IF(@counter < 10)
			SET @textAttrName = N'TextAttr0' +  CONVERT(varchar(2), @counter)
		ELSE
			SET @textAttrName = N'TextAttr' +  CONVERT(varchar(2), @counter)

		IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = @textAttrName)
		BEGIN
			PRINT 'Update <' + @textAttrName + '> column size to varchar 100'
			SET @sqlRequest = 'ALTER TABLE [<TableName,sysname,>] ALTER COLUMN [' +
@textAttrName + '] varchar(100)'
			EXEC (@sqlRequest)
		END
		SET @counter = @counter + 1;
	END

END
/* ******************************************************** */
/* --  STEP 4: BUILD INDEX ON MODIFIED TABLE IF REQUIRED  -- */
/* ******************************************************** */

IF @NeedsIndexRebuild = 1
BEGIN
	PRINT 'STEP 4 : Build primary index on modified table'
	ALTER TABLE [<TableName,sysname,>] ADD  CONSTRAINT
[IX_LOG_<TableName,sysname,>_PRIMARY] PRIMARY KEY CLUSTERED
	(
		[Chrono] ASC,
		[LogList] ASC,
		[EvtNumber] ASC,
		[Name] ASC,
		[Value] ASC,
		[Quality] ASC
	)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON
[PRIMARY]
END


/* ******************************************** */
/* --------- STEP 5 : UPDATE DB VERSION --------- */
/* ******************************************** */

PRINT 'Update database version'
UPDATE AICommonParameters
SET DbVersion = @DBVersionToReach

SET @CurrentDBVersion = (SELECT TOP 1 DbVersion FROM AICommonParameters)
PRINT 'Updated successfuly to version : ' + CONVERT(varchar(1), @CurrentDBVersion)
PRINT 'Update finished'
```

4.3 Upgrade the Trend Table

```
-- UpgradeTrendTable.sql

USE [<DatabaseName, sysname,>]

/*=============================================================================
  |                       BEFORE EXECUTING THIS SCRIPT                        |
  |                                                                           |
  | Use this template file to manually update an HDS TREND table.             |
  | Do not use this template file to update an HDS LOG table.                 |
  |                                                                           |
  | 1) Open this file with Sql Server management Studio (SSMS)                |
  | 2) Use 'specify values for template parameters ' as following             |
  |    Menu : Query -> Specify values for template parameters (Ctrl+Shift+M)  |
  | 3) Enter DatabaseName and TableName to update the script                  |
  | 4) Click on Execute as following :                                        |
  |    Menu : Query -> Execute (F5)                                           |
  | 5) Wait 'Update finished' message                                         |
   ============================================================================*/

/* ******************************************** */
/* --- STEP 1 : RETREIVE DB VERSION NUMBER    --- */
/* ******************************************** */
PRINT 'Retreive datebase version'
DECLARE @CurrentDBVersion INT;
DECLARE @DBVersionToReach INT;
DECLARE @NeedsIndexRebuild BIT;

SET @CurrentDBVersion = (SELECT TOP 1 DbVersion FROM AICommonParameters)
SET @DBVersionToReach = 3

PRINT 'Current version : ' + CONVERT(varchar(1), @CurrentDBVersion)

/* --- remove this part bellow if one of your tables has already been set to the current
version --- */
IF @CurrentDBVersion >= @DBVersionToReach
BEGIN
PRINT 'Current table version is up to date'
PRINT 'No further treatment required'
PRINT 'Update finished'
RETURN
END
/* --- remove this part above if one of your tables has already been set to the current
version --- */

IF @CurrentDBVersion = 1
BEGIN
      SET @NeedsIndexRebuild = 1;
      PRINT 'Version compatibility needs to rebuild the index to properly process'
      PRINT 'This task may take a while'
END

/* ****************************************************** */
/* --- STEP 2 : DROP INDEX ON TABLE TO UPDATE IF REQUIRED --- */
/* ****************************************************** */

IF @NeedsIndexRebuild = 1
BEGIN
      PRINT 'Drop index on table to update'
```

```
      ALTER TABLE <TableName,sysname,>
      DROP CONSTRAINT IX_TREND_<TableName,sysname,>_PRIMARY
END
/* ******************************************** */
/* -------- STEP 3 : UPDATE COLUMNS SIZE -------- */
/* ******************************************** */
PRINT 'STEP 2 : UPDATE COLUMNS SIZE'
IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'Name')
BEGIN
      PRINT 'Update <Name> column size to varchar 255'
      ALTER TABLE [<TableName,sysname,>]
      ALTER COLUMN [Name] varchar(255) NOT NULL
END

IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'Nature')
BEGIN
      PRINT 'Update <Nature> column size to varchar 100'
      ALTER TABLE [<TableName,sysname,>]
      ALTER COLUMN [Nature] varchar(100) NULL
END

IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'Domain')
BEGIN
      PRINT 'Update <Domain> column size to varchar 100'
      ALTER TABLE [<TableName,sysname,>]
      ALTER COLUMN [Domain] varchar(100) NULL
END

IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'Description')
BEGIN
      PRINT 'Update <Description> column size to varchar 255'
      ALTER TABLE [<TableName,sysname,>]
      ALTER COLUMN [Description] varchar(255) NULL
END

IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = 'AssocLabel')
BEGIN
      PRINT 'Update <AssocLabel> column size to varchar 255'
      ALTER TABLE [<TableName,sysname,>]
      ALTER COLUMN [AssocLabel] varchar(255) NULL
END

DECLARE @sqlRequest NVARCHAR(max)
DECLARE @counter INT;
DECLARE @textAttrName varchar(10)

SET @counter = 3;
WHILE (@counter < 17)
BEGIN
      IF(@counter < 10)
            SET @textAttrName = N'TextAttr0' +  CONVERT(varchar(2), @counter)
      ELSE
            SET @textAttrName = N'TextAttr' +  CONVERT(varchar(2), @counter)

      IF EXISTS (SELECT 1 FROM INFORMATION_SCHEMA.COLUMNS  WHERE
TABLE_NAME='<TableName,sysname,>' AND COLUMN_NAME = @textAttrName)
      BEGIN
            PRINT 'Update <' + @textAttrName + '> column size to varchar 100'
```

```
            SET @sqlRequest = 'ALTER TABLE [<TableName,sysname,>] ALTER COLUMN [' +
@textAttrName + '] varchar(100)'
            EXEC (@sqlRequest)
      END
      SET @counter = @counter + 1;
END

/* ******************************************************** */
/* --  STEP 4: BUILD INDEX ON MODIFIED TABLE IF REQUIRED  -- */
/* ******************************************************** */

IF @NeedsIndexRebuild = 1
BEGIN
      PRINT 'STEP 4 : Build primary index on modified table'
ALTER TABLE [<TableName,sysname,>] ADD  CONSTRAINT
[IX_TREND_<TableName,sysname,>_PRIMARY] PRIMARY KEY NONCLUSTERED
(
      [Name] ASC,
      [Chrono] ASC,
      [Value] ASC,
      [Quality] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON
[PRIMARY]
END

/* ********************************************* */
/* --------- STEP 5 : UPDATE DB VERSION --------- */
/* ********************************************* */

PRINT 'Update database version'
UPDATE AICommonParameters
SET DbVersion = @DBVersionToReach

SET @CurrentDBVersion = (SELECT TOP 1 DbVersion FROM AICommonParameters)
PRINT 'Updated successfuly to version : ' + CONVERT(varchar(1), @CurrentDBVersion)
PRINT 'Update finished'
PRINT 'UPDATE FINISH'
```

4.4 Step 5 to execute after all tables has been modified

```
/* ********************************************* */
/* --------- STEP 5 : UPDATE DB VERSION --------- */
/* ********************************************* */

PRINT 'Update database version'
UPDATE AICommonParameters
SET DbVersion = 3

SET @CurrentDBVersion = (SELECT TOP 1 DbVersion FROM AICommonParameters)
PRINT 'Updated successfuly to version : ' + CONVERT(varchar(1), @CurrentDBVersion)
PRINT 'Update finished'
PRINT 'UPDATE FINISH'
```

ARC Informatique Headquarters and Paris office
2 avenue de la Cristallerie 92310
Sèvres - France

tel + 33 1 41 14 36 00
fax + 33 1 46 23 86 02
hotline +33 1 41 14 36 25
arcnews@arcinfo.com
www.pcvuesolutions.com

ISO 9001 and ISO 14001 certified

ARC Informatique – Update Database_EN_2016

Publication number: AT-2016-01-04