www.asix.com.pl

AsReport –

Asix System Reporting Functionality
Based on the Microsoft Reporting Services

Doc. No. ENP9100
Version: 2017-07-03

ASKOM

ASKOM Sp. z o. o., ul. Józefa Sowińskiego 13, 44-121 Gliwice,
tel. +48 32 3018100, fax +48 32 3018101,

http://www.askom.com.pl, e-mail: office@askom.com.pl

1. Introduction

The 6th version of Asix system has been enriched in the reporting system based on Microsoft Reporting Services. **The AsRaport reporting system is supported both by the classic Assix version and the Asix.Evo version.**

Microsoft® SQL Server™ Reporting Services is a comprehensive server platform designed to meet a broad range of enterprise-wide reporting-related needs. The Reporting Services (in fact a component of the SQL Server 2008 database) enable report creation from various data sources; report environment management (planning the moment of report generation/subscribing/access control), as well as delivering reports to the users in the format and the way most convenient for them. The delivery methods include e-mail subscription and embedding the reports in business applications and/or Web portals.

In order **to be able to cooperate with the Reporting Services reporting platform**, the Asix has been extended by the following components:

- AsRaport data server built in AsixEvo.exe (in Asix.Evo) / Askom.Data.Host program (in classic Asix)
- AsRaport query editor
- AsixConnect database (run on a Microsoft SQL server)
- independent alarm archiver to store alarms in Microsoft SQL database.

The **AsixConnect database** is a key element. Its stored procedures share process data from Asix applications to the reporting environment. The **AsRaport** query editor plays a vital role in the report designing phase as the tool used to work out necessary SQL queries. Data from Asix applications retrieved by the queries are used by the **Report Builder** or the **Report Designer** client applications of the Reporting Services environment to generate final reports. The AsRaport program is a visual editor so that SQL queries may be constructed even by users not familiar with syntax of the Structured Query Language.

Report Builder is a tool that enables the workers without programming skills to construct report definitions for individual report generation ad hoc (on demand) by end users. The Reporting Services environment is similar to the Microsoft Office package and may be intuitively operated especially by persons acquainted with the Office Excel or Access program.

Report Designer is a very flexible and productive reporting environment for programmers and advanced report designers. It is a component of SQL Server Business Intelligence Development Studio of Business Intelligence programming environment based on the well known Microsoft Visual Studio programming interface. Simple reports may be quickly and easily developed with the help of the Report Wizard, more sophisticated ones may be worked out in visual development environment of the Report Designer platform.

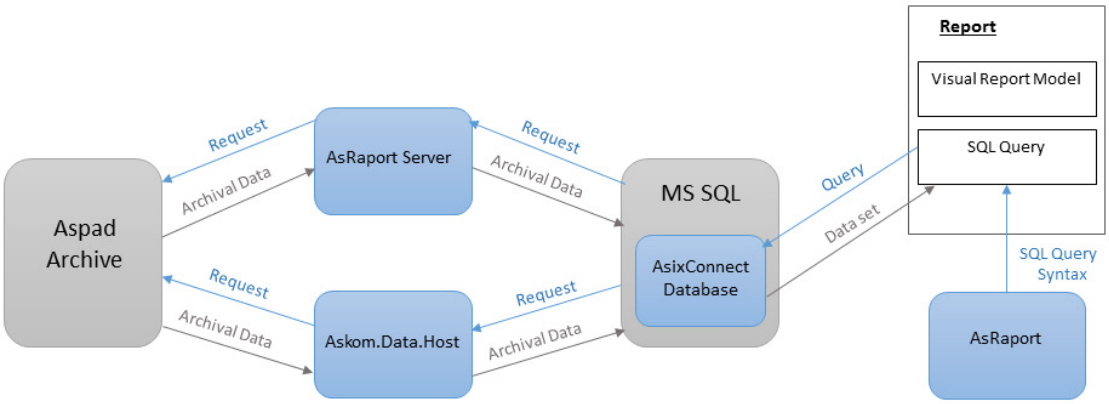

Fig. Architecture of the Interface Between the Asix System and the Reporting Services Environment.

Users who develop reports that are going to retrieve some data from Asix system applications simultaneously works in AsRaport interface and Report Builder/Designer interface.

The AsRaport and AsRaport server/Askom.Data.Host.exe programs must be first configured and the AsixConnect database must be created to get the Asix system ready for cooperation with the Reporting Services environment.

Design works on each new report form starts from defining syntax of SQL queries posted to database of stored procedures in order to retrieve source data, on which the reports are to be based; the AsRaport query editor is used to that end. Next, report layout must be specified in the Report Builder/Designer program - it demands a layout template with pre-defined data sections (tables, arrays, charts) to be selected, selected data elements to be placed in the project view, and suitable filtering criteria to retrieve source data for the reports to be defined. Report Builder/Designer may also be used to add new computational fields to the report form, almost arbitrarily format appearance of the report, and to view/print/Web-publish the report. A completed report form is stored on the Reporting Services central report server. The server shares the stored report forms throughout the entire enterprise using various methods, including making them available to Internet browsers.

More information on generation of reports in SQL Server Reporting Services may be found on the Microsoft Webpage.

1.1 Software Requirements

To make the Asix system compatible with the Reporting Services environment by Microsoft, you need to install SQL Server first; the minimum required version is 2008 R2 (the latest version is recommended).

The Reporting Services require at least the SQL Express with Advanced Services version. It is recommended to operate Windows 10/Windows Server 2016 with SQL Server 2016.

2 Procedure for Configuring Reporting Environment AsRaport to Work with MS Reporting Services

When starting to design reports with the use of the Reporting Services, first install the MS SQL Server reporting environment, 2008 R2 being the minimum version required (the latest version is recommended) on the computer on which the reports will be designed. The Reporting Services require at least the SQL Express with Advanced Services version.

The next stage is to configure the AsRaport reporting environment so as to make it compatible with MS Reporting Services and with the Asix application process data. This stage requires carrying out operations described in later subchapters of this documentation.

2.1 AsixConnect Database Creation

AsixConnect database with use of the AsRaport server (or Askom.Data.Host program in a classic version of Asix) retrieve process data from Asix applications and make them available to the reports.

AsixConnect creation is the first step of the procedure to get the Asix system ready for reporting based on Microsoft Reporting Services. Database of AsixConnect can be created in two ways: using Asix Evo or using the classic Architect (dedicated to classic versions of Asix applications).

ATTENTION: AsixConnect database is indispensable if Asix application process data are to be used by the reporting system.

To create AsixConnect database with the use of Asix Evo:

1. Run the AsixEvo.exe program and open in that program the Asix application configuration file.
2. Run the AsRaport set-up program:

   AsixEvo.exe > *Stations Settings* > *AsReport* > **Configure AsixConnect Database**

3. Select MS SQL server name for the AsixConnect database and click the ***Create Database*** button. The operation is always done for the current user of Windows.

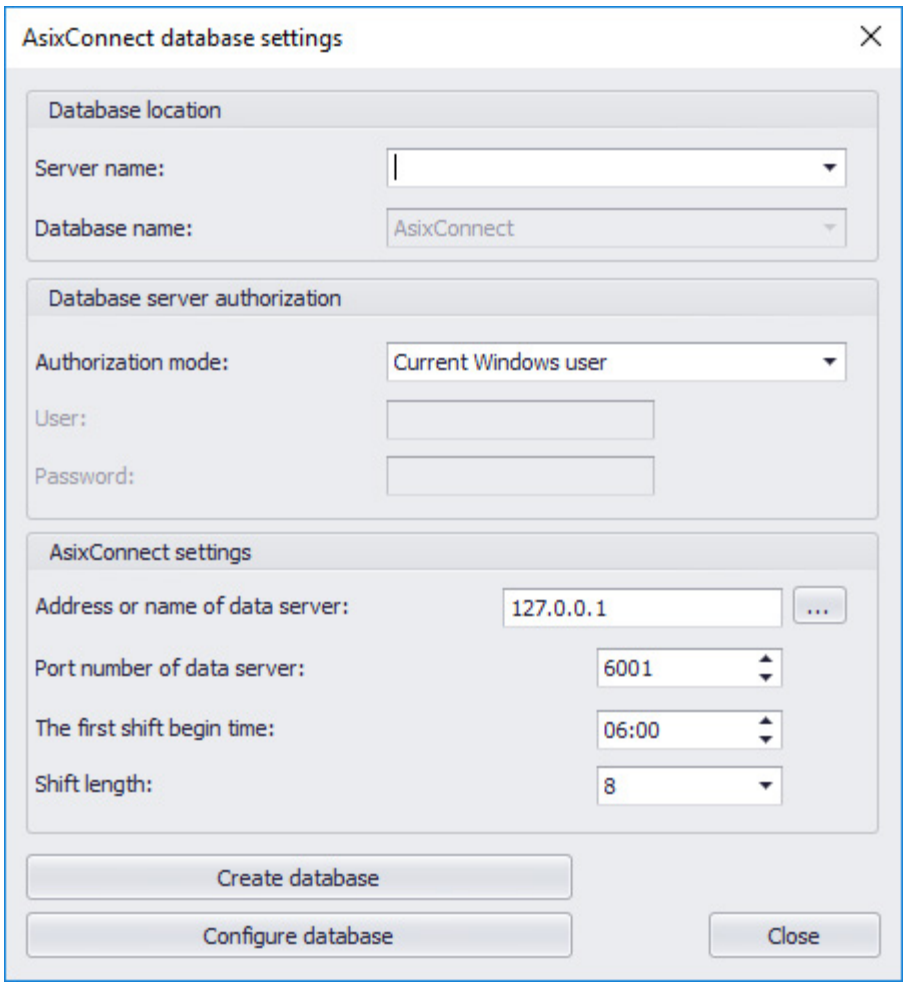

Fig. "AsixConnect database configuration" window.

To create AsixConnect database with the use of classic Architect:

1. Run the Architect program and open in that program the Asix application configuration file.
2. Run the AsRaport set-up program:

Architect > *Application* menu> ***Configure Reporting System AsRaport...***

3. On the ***AsixConnect Database/Location*** tab choose the name of MS SQL server and the name of AsixConnect database. You can also define the name of the SQL user authorised to create the database.

The AsixConnect database can be created both on a local and a remote server.

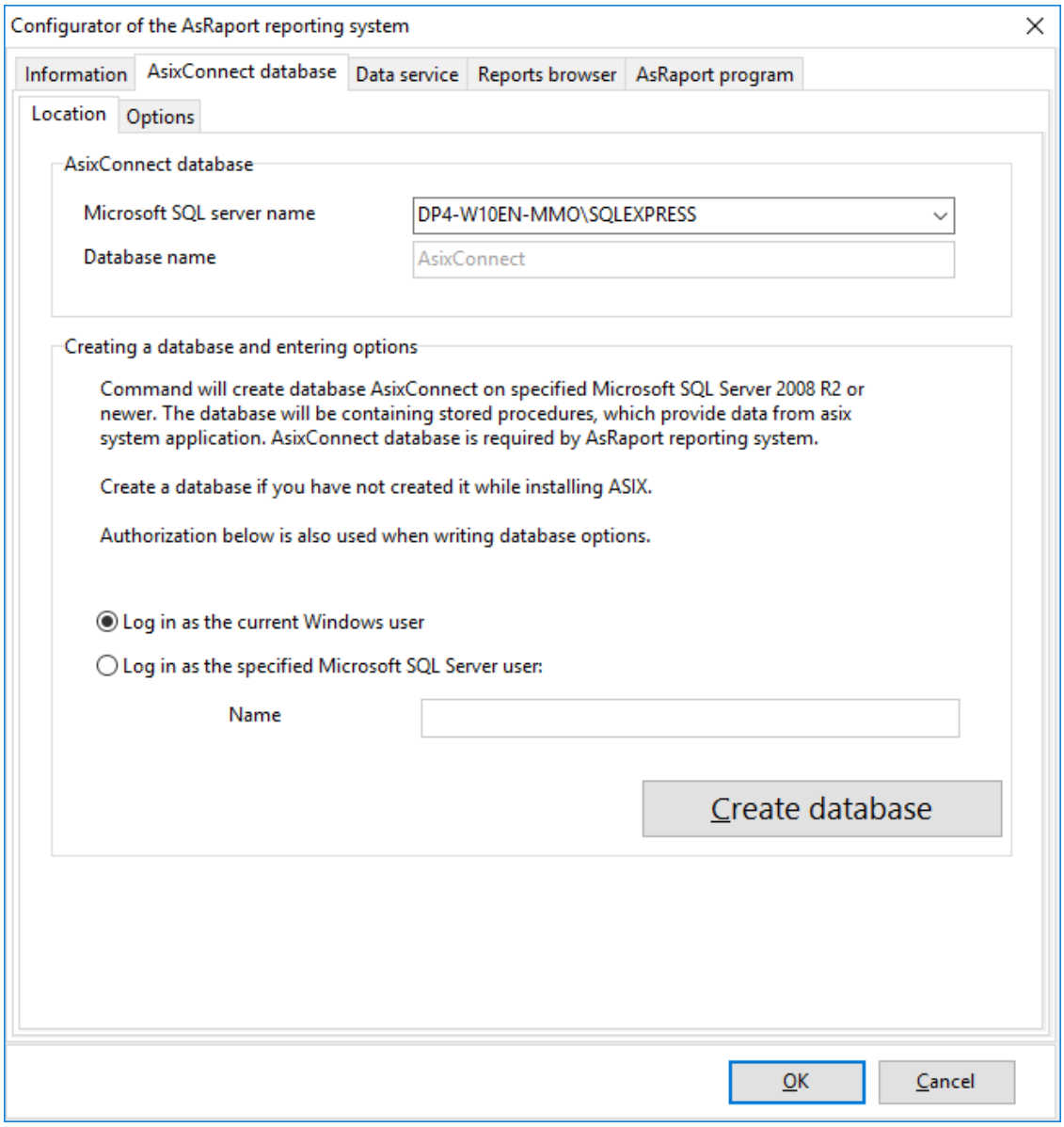

Fig.: 'Configurator of the AsRaport Reporting System' Window – AsixConnect Database/Location.

4. Click the button ***Create Database***.

2.2 Launch of Data Sharing Services for AsRaport Environment Reports

For Asix Evo applications:

1. Run the AsixEvo.exe program and open in that program the Asix application configuration file.
2. Run the AsRaport set-up program:

AsixEvo.exe > *Stations Settings* > *AsReport*

3. In the "*AsReport Data Server Settings*" enable the option ***Run Service for AsReport Reports.***

For Classic Asix applications:

1. Run the Architect program and open in that program the Asix application configuration file.
2. Run:

Architect > *Start Parameters* > Programs > Askom.Data.Host > ***Start Askom.Data.Host*** command.

2.3 Declaration of Report Server of MS Reporting Services

Declaration of Report Server of MS Reporting Services is used to open reports created in MS Reporting Services environment, directly from the Asix system application. These reports can then be viewed using the appropriate Asix system browsers (both in the classic and Evo versions).

To declare report server of MS Reporting Services for Asix Evo applications:

1. Run the AsixEvo.exe program and open in that program the Asix application configuration file.
2. Run the AsRaport set-up program:

   AsixEvo.exe > *Stations Settings* **>** *AsReport* **>** *AsReport Viewer Settings*

3. Enable the option ***Run AsReport Reports***.
4. Declare **Repporting Server Address** and select the to-be-used user authentication method.

   Two options are available:

   1. Windows Integrated Authorization.
   2. Standard Windows Authorization with User and Password.

Insert the full **address of the report site on the report server (Report Server Web Service URL)**, which can be found eg. using **Reporting Services Configuration Manager**. Exemplary address: http://192.168.1.100/ReportServer

Many addresses separated by semicolons are allowed. The browser connects to the first server with which manages to establish communication.

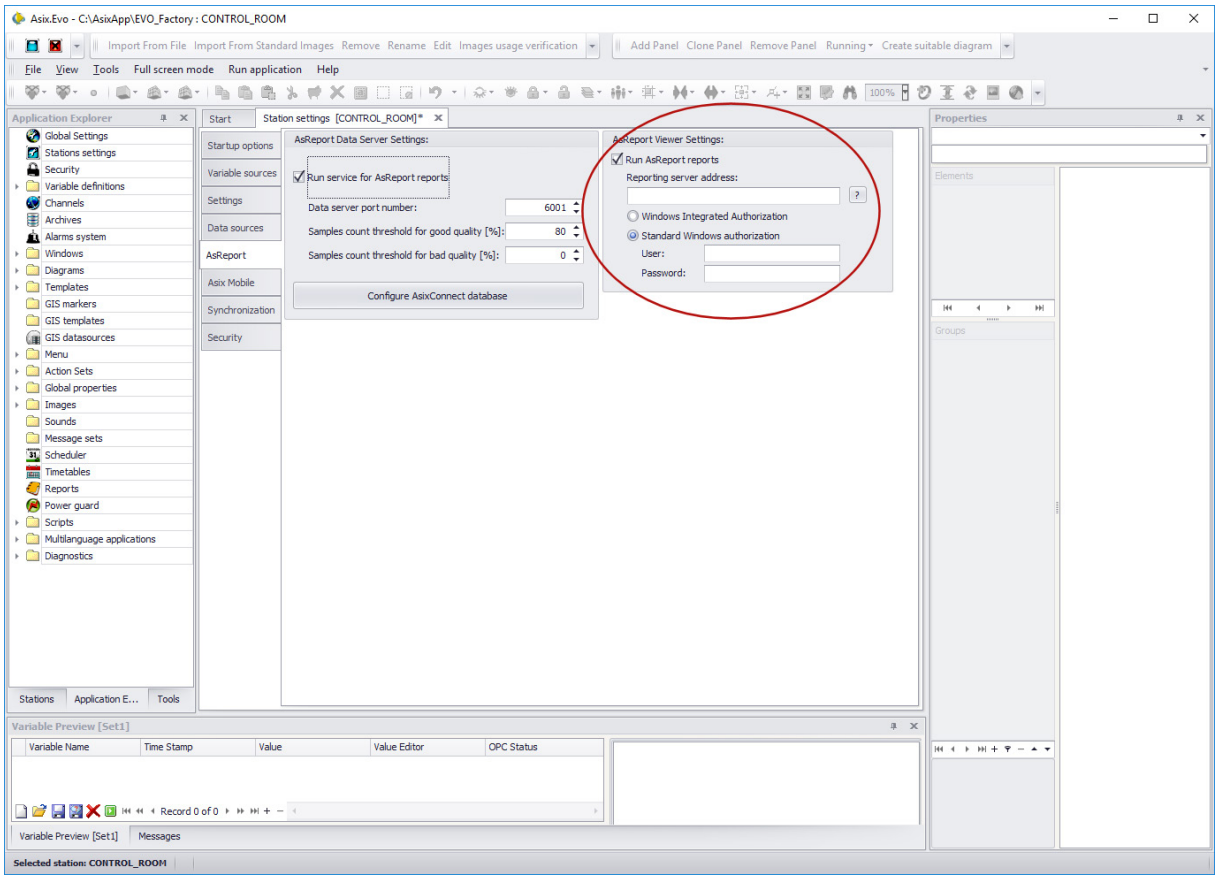

Fig.: 'AsReport Viewer Settings' Window.

To declare report server of MS Reporting Services for Classic Asix applications:

1. Run the Architect program and open in that program the Asix application configuration file.

2. Run the AsRaport set-up program:

Architect > *Application* menu**>** ***Configure Reporting System AsRaport...*** **>** ***Reports Browser*** tab

3. Declare **Report Server Web Service URL** (which can be found e.g. by means of the Reporting Services Configuration Manager program), name of the directory in the report server and select the to-be-used user authentication method.

Three options used to select method to authenticate users include:

1. Log-in as the current Windows user – use operating system user credentials
2. Log-in as the Windows user 'AskomInternal' – created by default when installing the Asix package
3. Log-in as the specified Windows user

4. Click ***OK***.

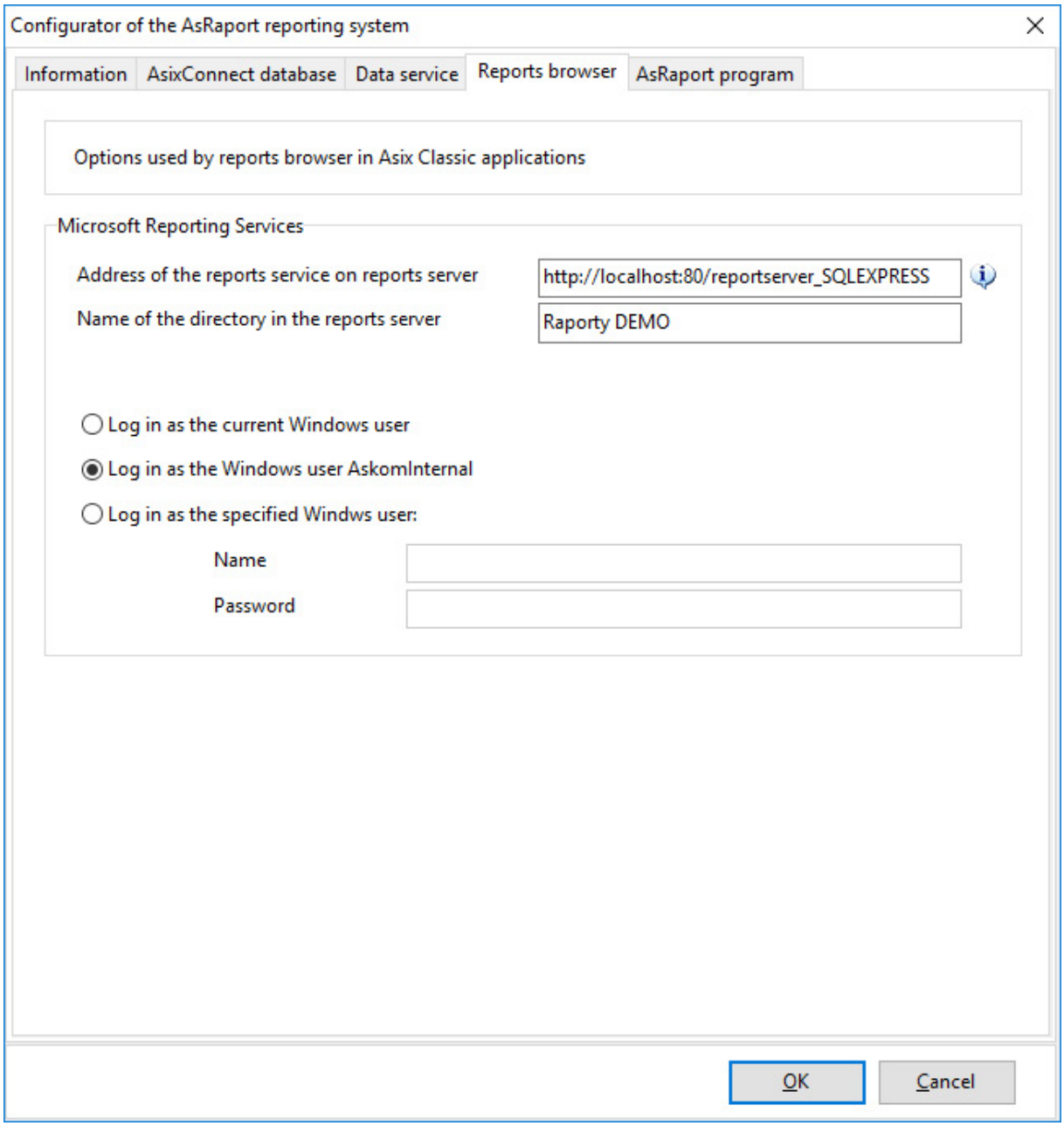

Fig.: *AsRaport reporting system configurator window – Reports browser.*

2.4 AsixConnect Database Configuration

To configure AsixConnect database with the use of Asix Evo:

1. Run the AsixEvo.exe program and open in that program the Asix application configuration file.

2. Run the AsRaport set-up program:

AsixEvo.exe > *Stations Settings* > *AsReport* > ***Configure AsixConnect Database***

3. Select MS SQL server name for the AsixConnect database and define the database server authorisation.

4. In the ***AsixConnect Settings*** group it is possible to:

- define a different location of the data server (by default, the data server and the AsixConnect database are located on the same computer);

- configure a shift system:

The First Shift Begin Time - start time of the first shift;

Shift Length - period of time declared in hours.

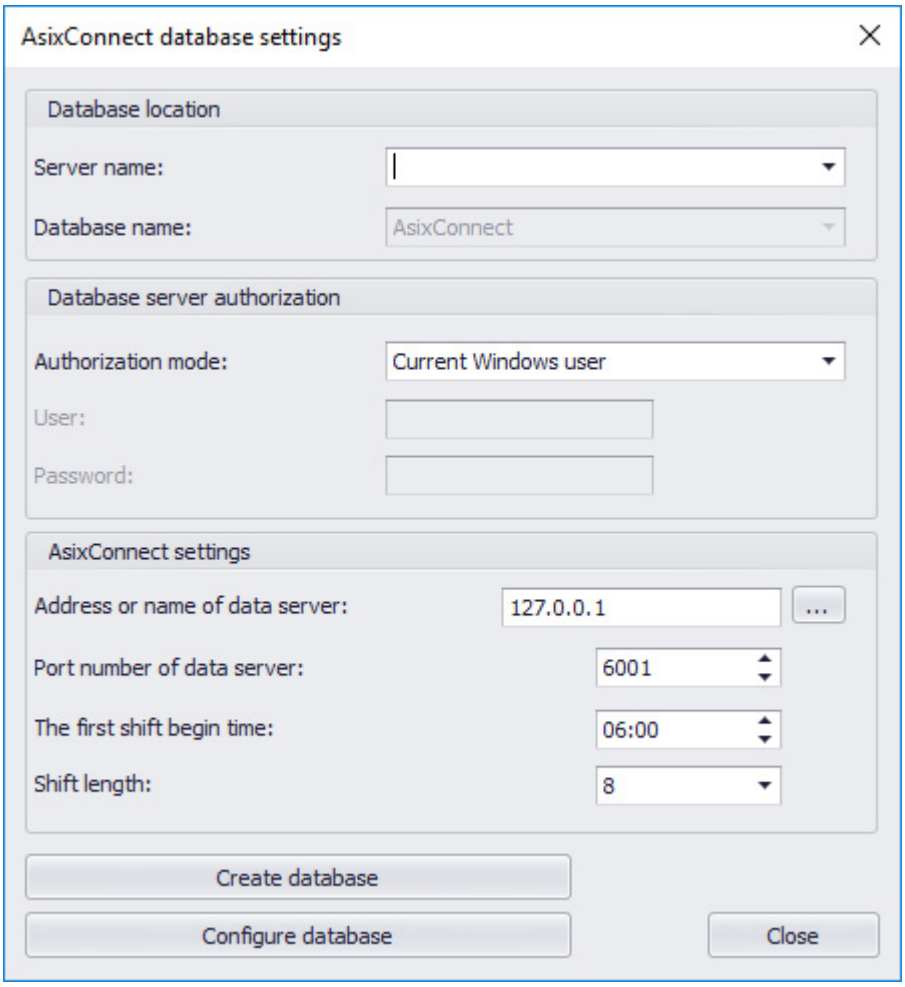

Fig. "AsixConnect database configuration" window.

5. Select the Configure database button.

To configure AsixConnect database with the use of classic Architect:

1. Run the Architect program and open in that program the Asix application configuration file.

2. Run the AsRaport set-up program:

Architekt > ***Application*** menu > ***Configure Reporting System AsRaport...***

3. On ***AsixConnect Database / Options*** tab there is the possibility to:

- define a different location of the Askom.Data.Host data server (by default, the Askom.Data.Host data server and the AsixConnect database are located on the same computer);

- declare ***Treatement of the Points with Uncertain Quality*** as points with a good quality or with a bad quality;

- declare ***Configuration of a Shift System***:

The Beginning of the First Shift – start time of the first shift;

The length of the Shift - period of time declared in hours.

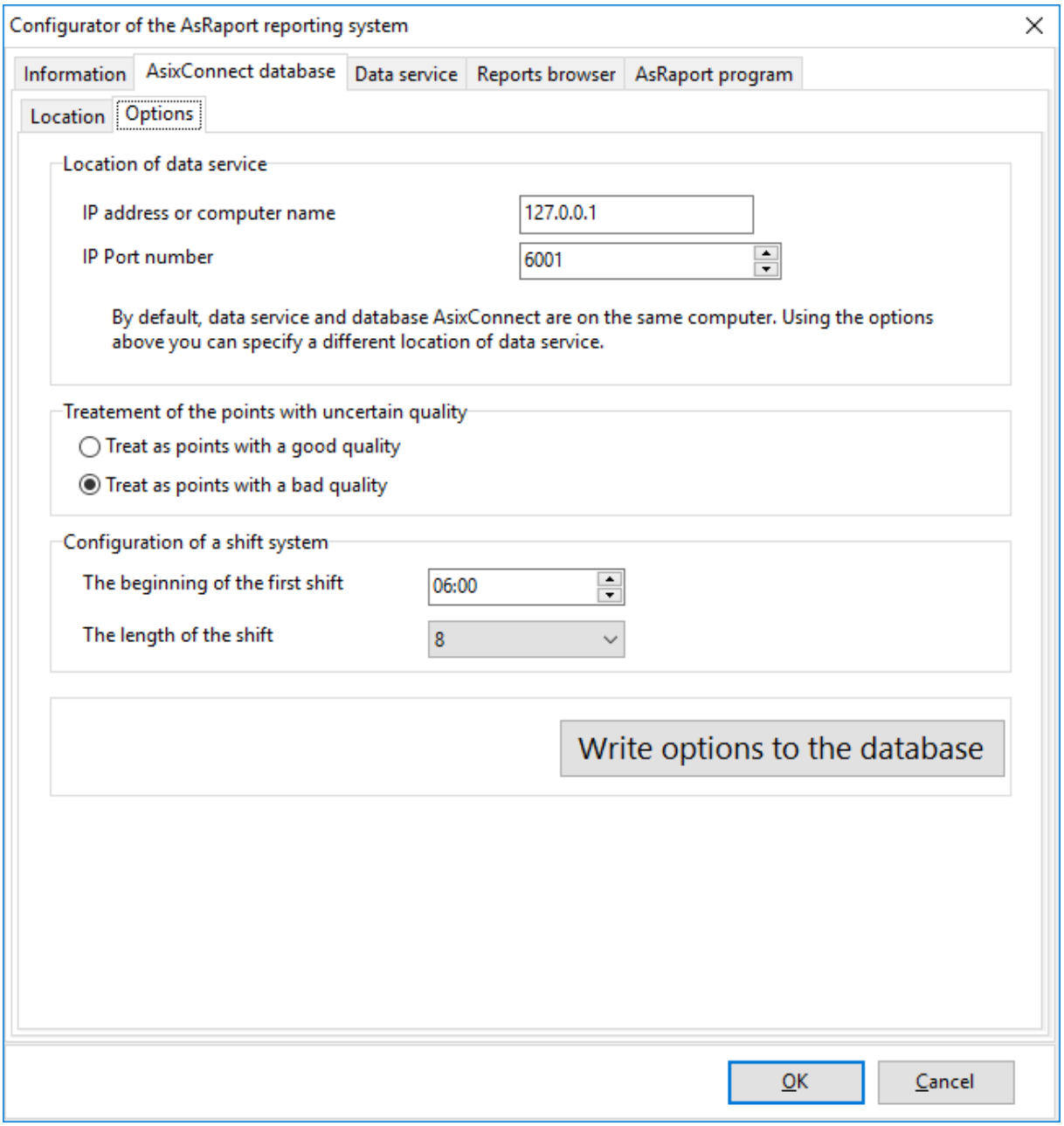

Fig.: AsRaport reporting system configurator window – AsixConnect database/Options.

4. These settings will be saved in the AsixConnect database by means of the ***Write Options to the Database*** button.

To declare quality thresholds for Asix Evo applications:

1. Run the AsixEvo.exe program and open in that program the Asix application configuration file.
2. Run the AsRaport set-up program:

AsixEvo.exe > *Stations Settings* **>** *AsReport* **>** *AsReport Data Server Settings*

3. Declare sample count treshold for good and bad quality*:*

Samples Count Threshold for Good Quality [%]- the aggregate has a good quality if a number of good samples in the interval is equal to or exceeds a predefined threshold of good quality.

Samples Count Threshold for Bad Quality [%] - the aggregate has a bad quality if a number of bad samples in the interval is equal to or exceeds a predefined threshold of bad quality.

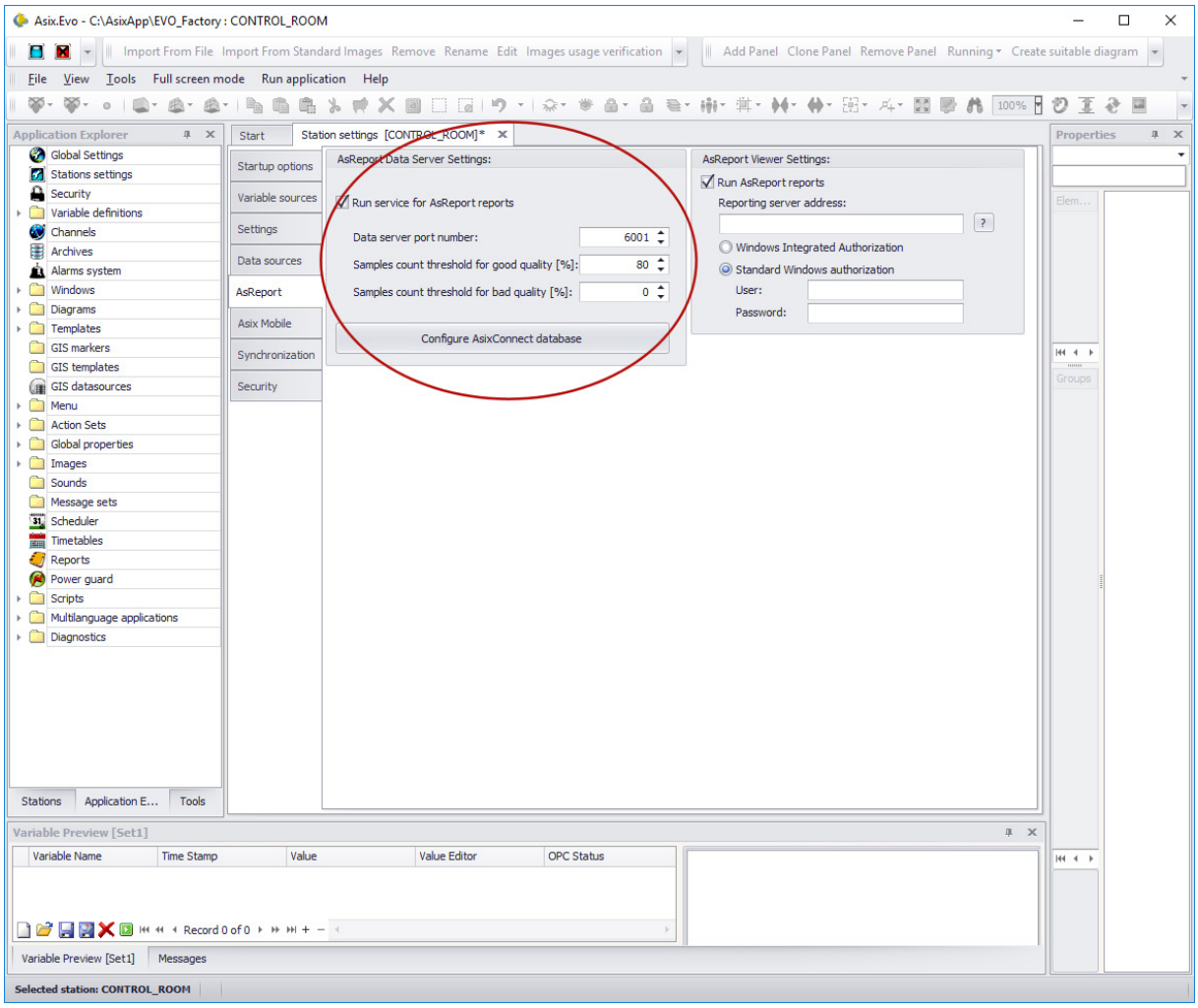

Fig. 'AsReport Data Server Settings' - Options of Quality Tresholds of Archival Data.

To declare quality thresholds for Classic Asix applications:

1. Run the Architect program and open in that program the Asix application configuration file.

2. Run the AsRaport set-up program:

Architect > ***Application*** menu **>** ***Configure Reporting System AsRaport...*** **>** ***Data Service*** tab

3. Declare threshold for good and bad quality*:*

Good Quality Threshold - the aggregate has a good quality if a number of good samples in the interval is equal to or exceeds a predefined threshold of good quality.

Bad Quality Threshold - the aggregate has a bad quality if a number of bad samples in the interval is equal to or exceeds a predefined threshold of bad quality.

4. Click ***OK***.

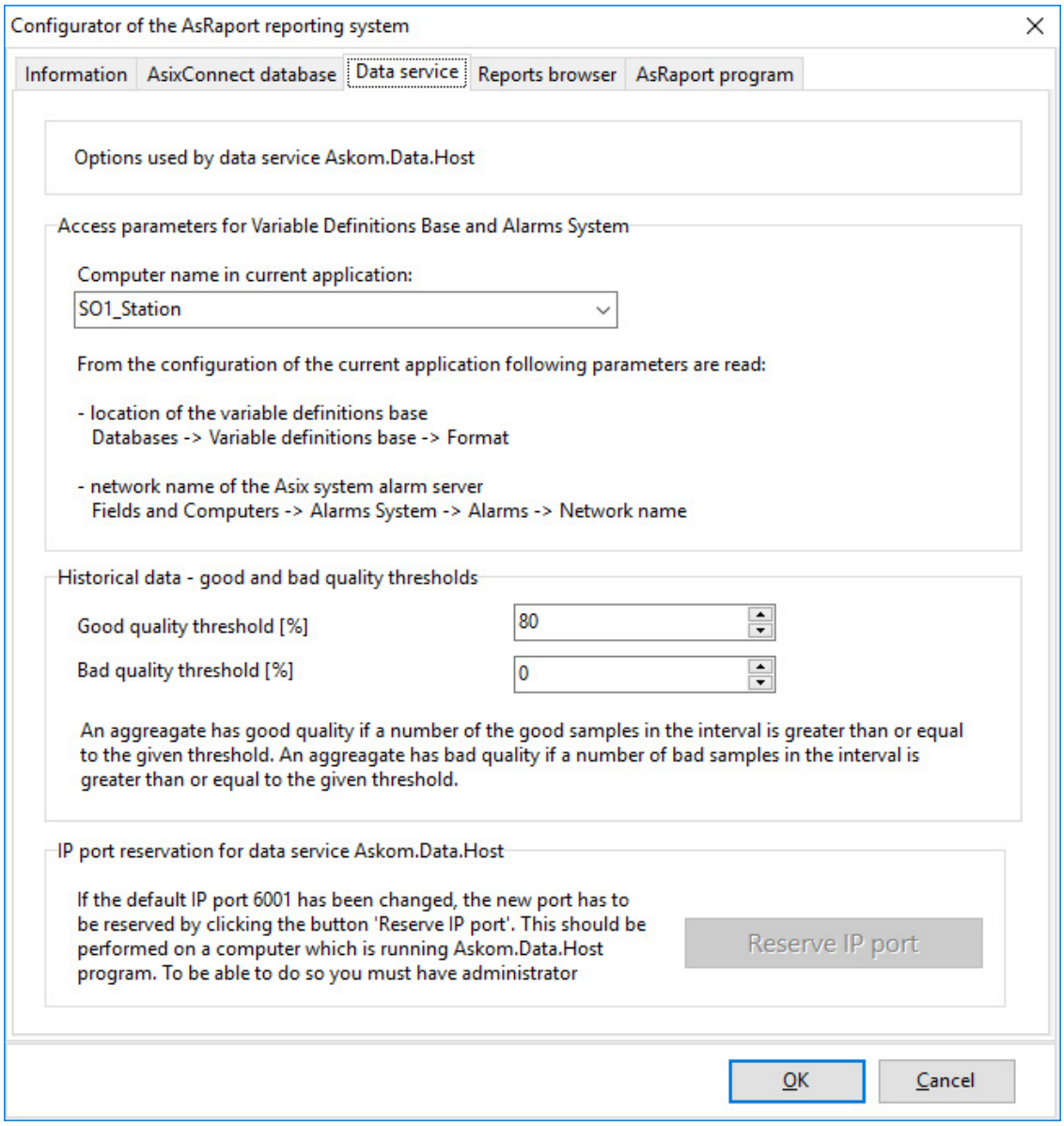

Fig. 'Configurator of the AsReport Reporting System' Window – Data Service.

3 AsixConnect SQL Queries Built in AsReport

The AsRaport query editor is run by the AsRaport.exe executable put by default by the Asix package installer into the package C:\Program Files\Askom\Asix main folder.

The ***Tools*** and ***View*** tabs at the top part of the main window of the editor display collections of tools. The remaining part of the window is consumed by ***Query1***, ***Query2***,... tabs used to define various queries to the database of process value archive and to the database of variable definitions.

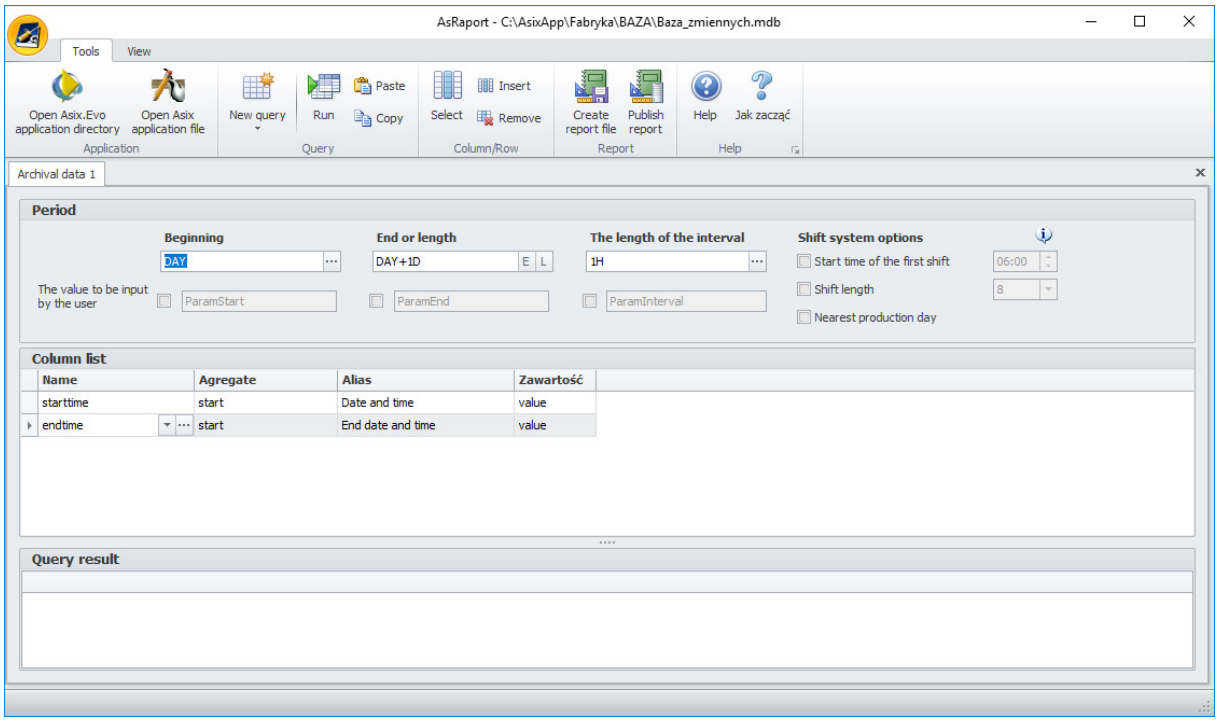

Fig: AsRaport – main window.

The Tools tab includes:

Open Asix.Evo application directory – open the Asix.Evo application configuration file (XML file).
Open Asix application file – the command oopens the XML configuration file of the Asix application, the classic version.

New query – Archival data – add a tab for a new query to the database of process value archive.
New query – Raw data – adds a new tab allowing you to build SQL queries to raw data.

New query – variable definitions database – add a tab for a new query to the database of variable definitions.
New query – Alarms – adds a new tab allowing you to build SQL queries to the alarm archive.

Run – display results of the so-far defined query.

Copy – copy query defined in the AsRaport window to the Windows Clipboard (to paste it later at the report development stage to the Report Builder/Designer query definition window).

Paste – paste a query from the Windows Clipboard to the AsRaport window.

Select – select process variable.

Insert – insert a new row to list of columns (in case of a query to the database of process value archive) or a new row to list of attributes/variables (in case of a query to the database of variable definitions).

Remove – remove the highlighted row from list of columns.

Tools of the ***View*** tab are used to select styles and colors used in the AsRaport window to control its appearance.

Each Query tab opened by the ***New query – archival data*** command displays the following elements:

Specifications of the ***Period*** to be covered by the query:

Beginning – beginning of the period

End or length – end or length of the period from within which the process variable data will be sourced

The length of the interval – interval between subsequently retrieved process variable values (the entire query period is divided into constant-length time intervals).

Shift system options*: *Start time of the first shift*, *Shift length*, *Nearest production day.

Column list – list of fields (report columns) to be presented in the report. Click in the new row in the ***Name*** column, then click the  button to drop down selection list of all pre-defined field names:

starttime – interval start time

endtime – interval end time

shiftnumber – operator shift number. Three shifts have been pre-defined: 6 am–2 pm (the 1st shift); 2 pm–10 pm (the 2nd shift); and 10 pm–6 am (the 3rd shift).

shiftdate – date when the given shift commenced.

starttimeutc - interval start time in the UTC time

endtimeutc - interval end time in the UTC time

For each query field an **Agregat** (average0, average, delta, end, max, min, range, start, Total, quality_bad, quality_good, qyality_uncertain, quality_good_duration, quality_uncertain_duration, quality_bad_duration, sum_up, sum_down, previous_known, last, root_mean_square, standard_deviation, total_last_known, average_last_known) and an **Alias** may be defined. The aggregate is calculated for each interval using all archived data belonging to that interval.

Query result – result of a query posted to the database of process value archive.

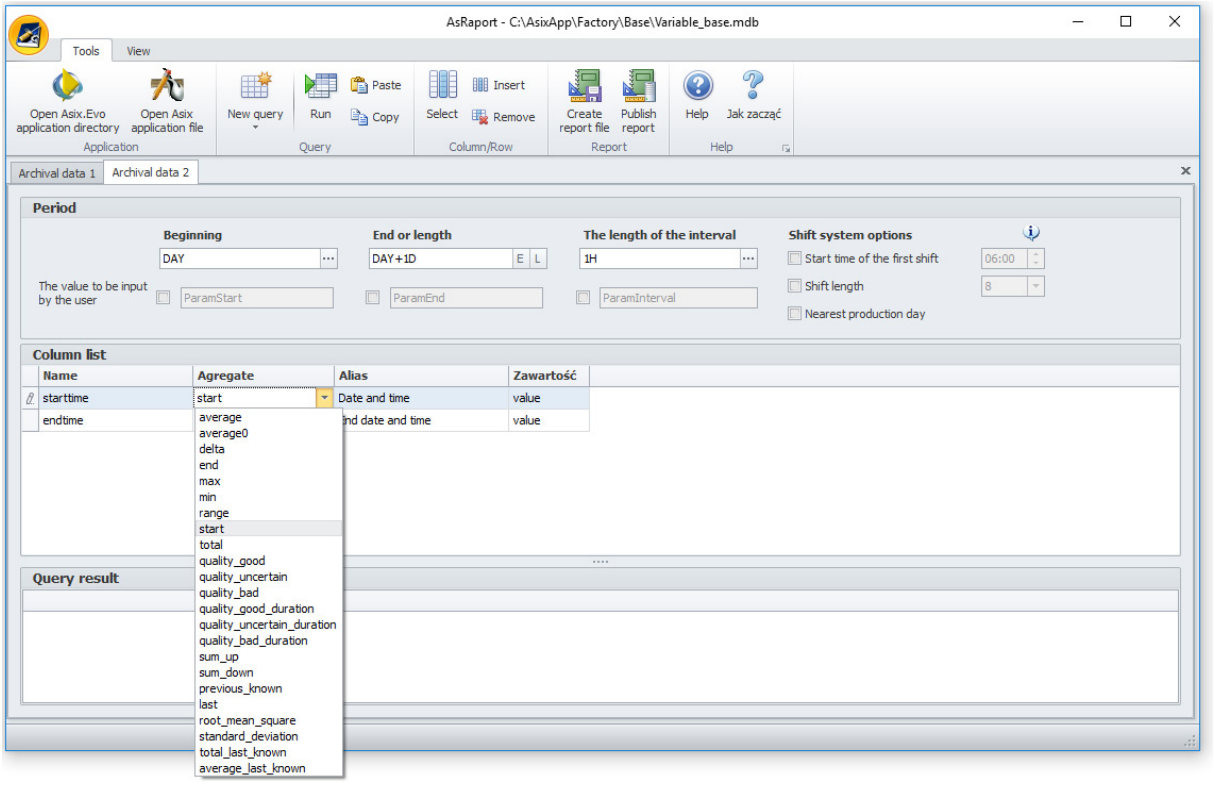

Fig. Declaration of Query Field for Inquiries to Archival Data.

Each ***Query*** opened by the ***New query – Raw data*** command displays the following elements:

Specifications of the ***Period*** to be covered by the query:

Beginning – beginning of the period

End or length – end or length of the period from within which the process variable data will be sourced

Shift system options*: *Start time of the first shift*, *Shift length*, *Nearest production day.

Column list – list of fields (report columns) to be presented in the report. Click in the new row in the ***Name*** column, then click the button to drop down selection list of all pre-defined field names:

time – sample time

shiftnumber – operator shift number. Three shifts have been pre-defined: 6 am–2 pm (the 1st shift); 2 pm–10 pm (the 2nd shift); and 10 pm–6 am (the 3rd shift).

shiftdate – date when the given shift commenced.

timeutc

For each query field an **Alias** may be defined.

Query result – result of a query posted to the variable database.

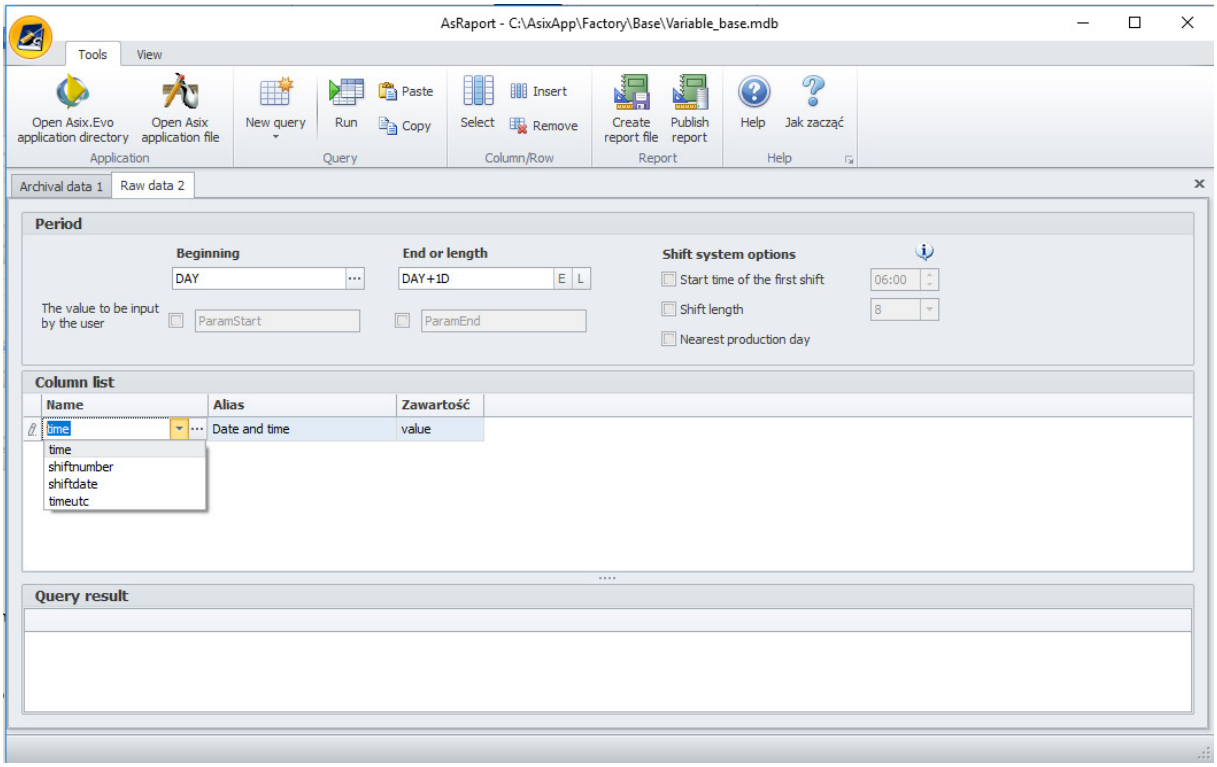

Fig. Declaration of Query Field for Inquiries to Raw Data.

Each ***Query*** opened by the ***New query - variable definitions database*** command displays the following elements:

Layout of the result table – attribute names are names of columns and row contains the values of selected attributes of one variable or variable names are names of columns and row contains the values of one attribute of selected variables.

Attribute list – list of attributes of variables to be presented in the report.

Variable list – list of names of variables to be presented in the report.

Query result – results of the query posted to the database of variable definitions.

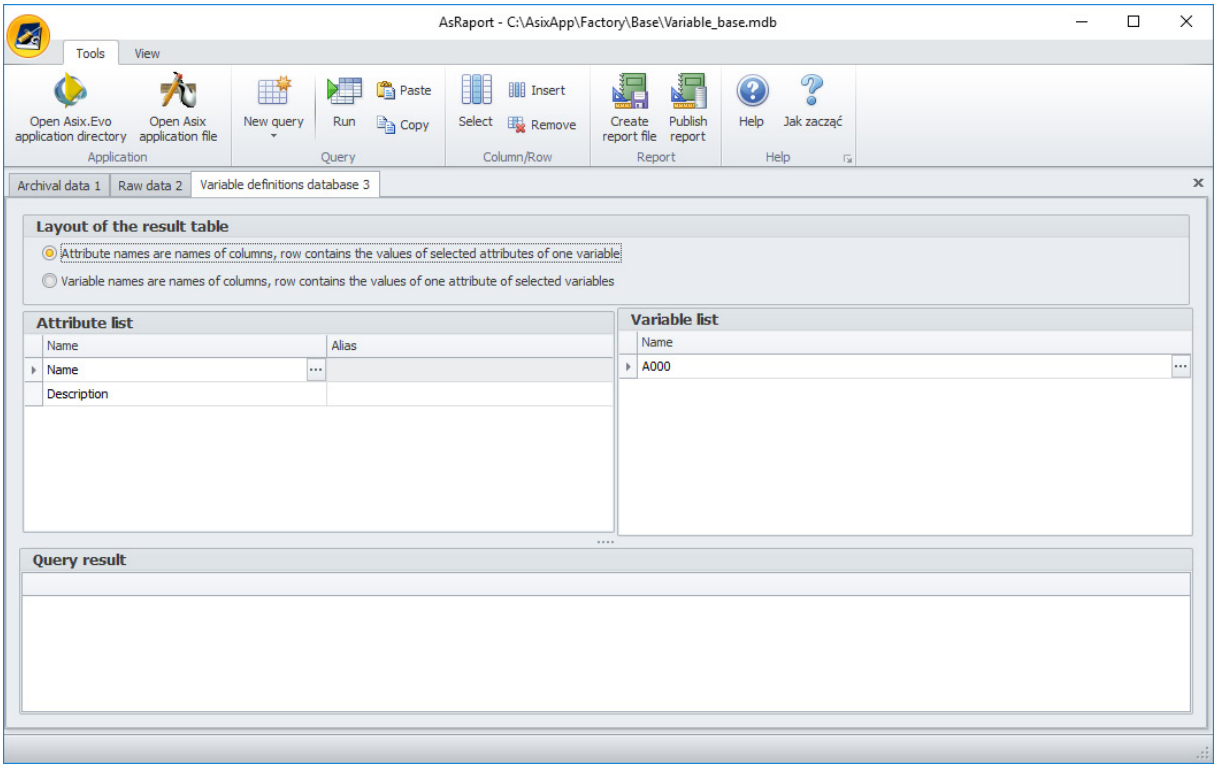

Fig. Declaration of Query Field for Inquiries to Variable Definition Database.

Each ***Query*** opened by the ***New query - Alarms*** command displays the following elements:

Data kind – defines the kind of data to be displayed: active periods for historic alarms or the total duration of the active periods of the historic alarms.

Alarms domain – it defines the alarm domain which unequivocally identifies a set of alarms, that is, the alarm archive, alarm definitions and detection strategies. This is an equivalent to the alarm resource name in the older classic Asix applications.

Alarm text language

Begin of period – beginning of the period from which the data will be sourced.

End of period – end of the period from which the data will be sourced.

Alarm identifier pattern - you can use wildcard characters of the SQL language, that is, '%' and '_'. Generally, this pattern should be consistent with the SQL language LIKE phrase syntax.

Alarm text pattern - you can use wildcard characters of the SQL language, that is, '%' and '_'. Generally, this pattern should be consistent with the SQL language LIKE phrase syntax.

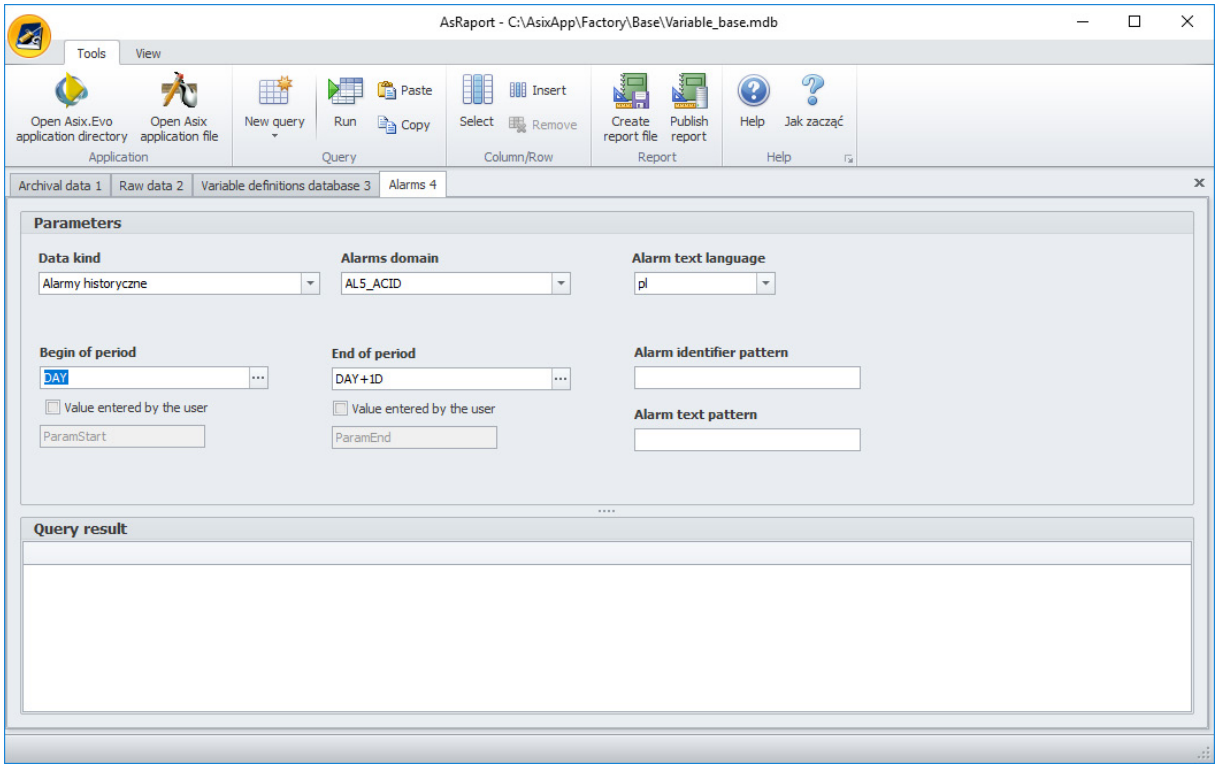

Fig. Declaration of Query Field for Inquiries to Alarm Archive.

3.1 Authorisation of Access to AsixConnect Database for AsRaport program:

The AsReport program access to the AsixConnect database must be authorised for the data to be read:

(**The MS SQL server access method can only be configured through the Architect program.**)

 **To configure access to the AsixConnect database for the AsRaport program:**

1. Run the AsixEvo.exe program and open Asix application configuration file in it. Go to *Stations settings* > *Data sources* > *Connect via Asix Data Server* > data server file edition button by means of the Architekt button. For the classic Asix version: Run the Architekt program and open the Asix application configuration file in it.

2. Run the AsRaport reporting system configurator:

Architect > *Application* menu > ***Configure Reporting System AsRaport...***

3. In the ***AsRaport program*** tab, specify the logging in method and confirm your choice with the ***OK*** button:

There are three ways available to verify the AsixConnect database users:

1. ***Log in as current Windows user*** – logging in of the current user of the Windows operating system

2. ***Log in as a user of Microsoft SQL Server named '_asix_internal'*** – created by default when installing the Asix package (such user can be created on conditon that the MS SQL server has been pre-installed before starting the installation of the Asix package)

3. ***Log in as the specified user of Microsoft SQL Server***.

3.2 Data Retrieval – AsixConnect Stored Procedures

The AsRaport query visual editor may assist users who are not familiar with syntax of the SQL language. However, some users may prefer to construct the queries – including queries to alarm archives – on their own. Time-related parameters of the query (periodStart, periodEnd, resampleInterval, periodLen) may be specified in local time, or in the OPC time format (see the 3.2.1 Date/Time/Period OPC Format section below). Retrieval of archived process data, data from variable definition database, alarms from the SQL archive, and alarms from file archives is discussed in subsequent sections below.

3.2.1 Date/Time/Period OPC Format

The OPC time format

The syntax:
keyword +/- offset number +/- offset number …

'keyword': NOW, MINUTE, HOUR, DAY, WEEK, MONTH, YEAR
'offset': S, M, H, D, W, MO, Y

Examples: DAY-1D, YEAR+1MO

The OPC duration format

The syntax:
+/- offset number +/- offset number …
It doesn't contain 'keyword'
'offset': S, M, H, D, W, MO, Y

Examples: 5M, 1H, 12H, 1MO, 1D+12h

3.2.2 Retrieval of Archived Data

ReadProcessed *periodStart, periodEnd, resampleInterval, column1, column2, column3, …*

Time period specification:
periodStart, periodEnd: date, for example: '2009-1-1 0:0:0‘
resampleInterval: number of seconds
Note: the date and the time period available in the OPC format.

column1, column2, ... - names of variables from the variable definition database; the function may return not more than 18 columns.

ReadProcessedL *periodStart, periodEnd, resampleInterval, columnList*

Time period specification:

periodStart, periodEnd: date, for example: '2009-1-1 0:0:0‘

resampleInterval: number of seconds

Note: the date and the time period available in the OPC format.

columnList - is a text string containing all column names separated with comas.

Procedura ReadProcessed - columns with variable values

- Variable name: ‘A000’
- As before + alias: ‘A000 as [Flue gases temp.]’
- Aggregate and variable name: ‘avg(A000)’
- As before + alias: ‘avg(A000) as [Averaged flue gases temp.]’
- Quality and name of the variable: ‘quality(A000)’
- Quality, aggregate and name of the variable: ‘quality(avg(A000))’

Procedura ReadProcessed - time columns:

- stamp time of interval start – ‘starttime’
- stamp time of interval end – ‘endtime’
- date of shift start – ‘shiftdate’
- number of shift – ‘shiftnumber’

- Timestamp for the interval start time in the UTC time - ‘starttimeutc’
- Timestamp for the interval end time in the UTC time - ‘endtimeutc’

3.2.3. Retrieval of Raw Data

ReadRaw *periodStart, periodEnd, column1, column2, column3, …*

Time period specification:

periodStart, periodEnd: date, for example: '2009-1-1 0:0:0‘

Note: the date and the time period available in the OPC format.

column1, column2, ... - names of variables from the variable definition database; the function may return not more than 18 columns.

ReadRawL *periodStart, periodEnd, columnList*

Time period specification:

periodStart, periodEnd: date, for example: '2009-1-1 0:0:0‘

Note: the date and the time period available in the OPC format.

columnList - is a text string containing all column names separated with comas.

Procedura ReadRaw - columns with variable values

- Variable name: ‘A000’
- As before + alias: ‘A000 as [Flue gases temp.]’
- Quality and name of the variable: ‘quality(A000)’

Procedura ReadRaw - time columns

- Stamp time - ‘time’
- date of shift start - ‘shiftdate’
- number of shift - ‘shiftnumber’
- Timestamp in the UTC time - ‘timeutc’

3.2.4 Retrieval of Data from Variable Definition Database

Retrieve data from the database of variable definitions:

ReadAttributes *variableNames, column1, column2, column3*, …

variableNames: the list of variable names, for example:‘A000, A004, A008’

column1, column2, ...: attributes from the variable definition database (for the current list of attributes, see: Architekt.chm/pdf, Attachment 1)

Retrieve and transpose data from the database of variable definitions

ReadAttributesTranspose *attributeNames, column1, column2, column3, …*

attributeNames: the list of variable attributes, for example: ‘Name, Description, Unit’

column1, column2, ...: attributes from the variable definition database (for the current list of attributes, see: Architekt.chm/pdf, Attachment 1)

3.2.5 Retrieval of Historical Alarms (SQL Archive)

Retrieve historical alarms from the SQL archive:

ReadProcessedHistoricalAlarms resourceName, periodStart, periodEnd, alarmIdPattern, alarmTextPattern, alarmTextLang

Only alarms that were initiated within the specified period will be retrieved.

Alarm start/end/confirmation date, ID, text and duration will be retrieved.

resourceName: nazwa domeny alarmów aplikacji Asix Evo

periodStart, periodEnd: time period in a local format or in the OPC format

alarmIdPattern: a text for filtering the range of the returned alarms by the alarm identifier. It can be:

- Alarm identifier – only alarms with a specified identifier are returned
- List of alarm identifiers – only alarms with an identifier included in the list are returned; the list should be enclosed in the quotation marks
- Identifier pattern - only alarms with an identifier matching the pattern are returned; you can use the SQL language wildcard characters, that is, '%' and '_'. Generally, this pattern should be consistent with the SQL language LIKE phrase syntax

  1,2,3,

  1-3

alarmTextPattern: a text for filtering the range of the returned alarms by the alarm text. It can be:

- Alarm text – only alarms with a specified alarm text are returned
- Text pattern - only alarms with an alarm text matching the pattern are returned; you can use the SQL language wildcard characters, that is, '%' and '_'. Generally, this pattern should be consistent with the SQL language LIKE phrase syntax

  <text>

  alarmTextLang: alarm text language identifier (Polish language – 'pl')

Reading of historical alarms – active periods

ReadProcessedHistoricalAlarms_ActivePeriods *resourceName*, *periodStart*, *periodEnd*, *alarmIdPattern*, *alarmTextPattern*, *alarmTextLang*

Alarms which were active during the specified period of time are returned.
Each alarm start and end date which is beyond the specified period is narrowed to the specified time period.

Reading of historical alarms – total length of active periods

ReadProcessedHistoricalAlarms_ActivePeriodTotal ***resourceName*****,** ***periodStart*****,** ***periodEnd*****,** ***alarmId*****,** ***alarmTextPattern*****,** ***alarmTextLang***

The length of the period during which the alarm was active during the specified period of time is returned.

4 Designing Reports in Report Builder

Basic steps of the procedure that must be performed to develop a report utilizing some values of process variables/variable definitions retrieved from Asix system applications are described below using an example of the “Factory” demo application delivered with the Asix package.

Reporting Services from the SQL Server 2008 Service Pack 1 version Express have been used and Report Builder version 3.0 PL. For detailed instructions see documentation of the Report Builder program.

Main window of the Report Builder program in version 2 is shown in the figure below.

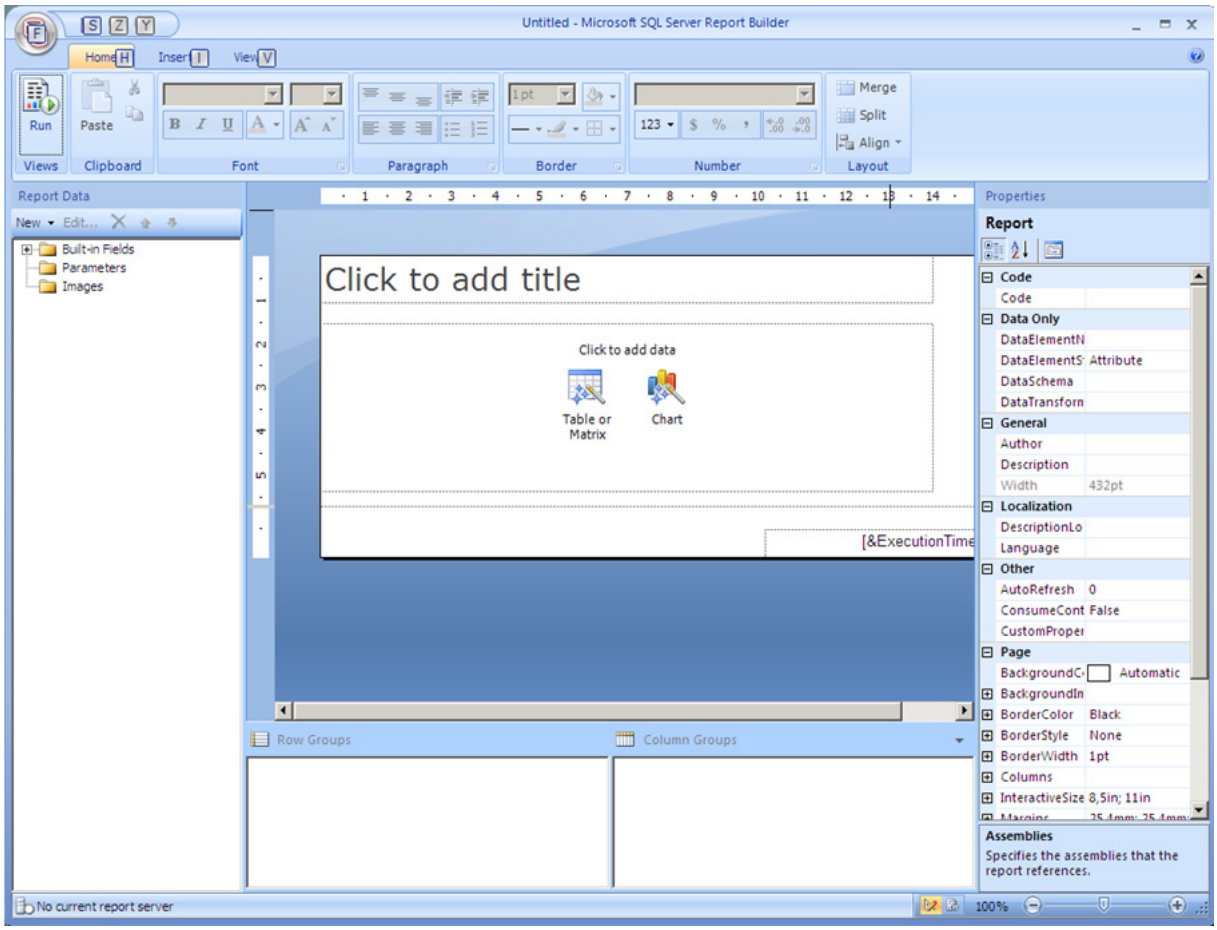

Fig. Report Builder - main window.

Form used to define new reports contains by default report contents, page footnote, and the following elements:

- text box with the “click to add title” prompt
- link to the table/matrix wizard and link to the chart wizard
- the embedded [&ExecutionTime] field in the footnote (to add report execution date/time to each page of the report at page bottom).

4.1 Create a New Report

Click the ***Start*** button and select the ***Programy/Microsoft SQL Server 2008 Report Builder/Report Builder 2.0*** option to create a new report. The displayed Report Builder window will show new report in the project view.

4.2 Declare Data Source/Layout/Format (Wizard)

The first step is to select source of the to-be-reported data and to indicate the format, in which the reported data are to be arranged. Data presented in examples below are arranged in tables.

Table may be added to the report using the table wizard. The wizard helps to define links to data sources, to create query specifying which data are to be reported, to organize data in some groups, and to add summary rows (in which data contained within the group in question are aggregated).

1. Click the ***Design*** button to switch to the project view.

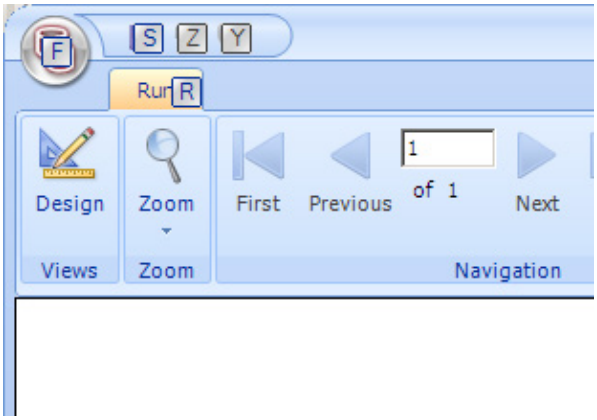

Fig. Design button to switch to the project view.

2. Click the ***Table or Matrix*** icon within the ***Click to add title*** area.

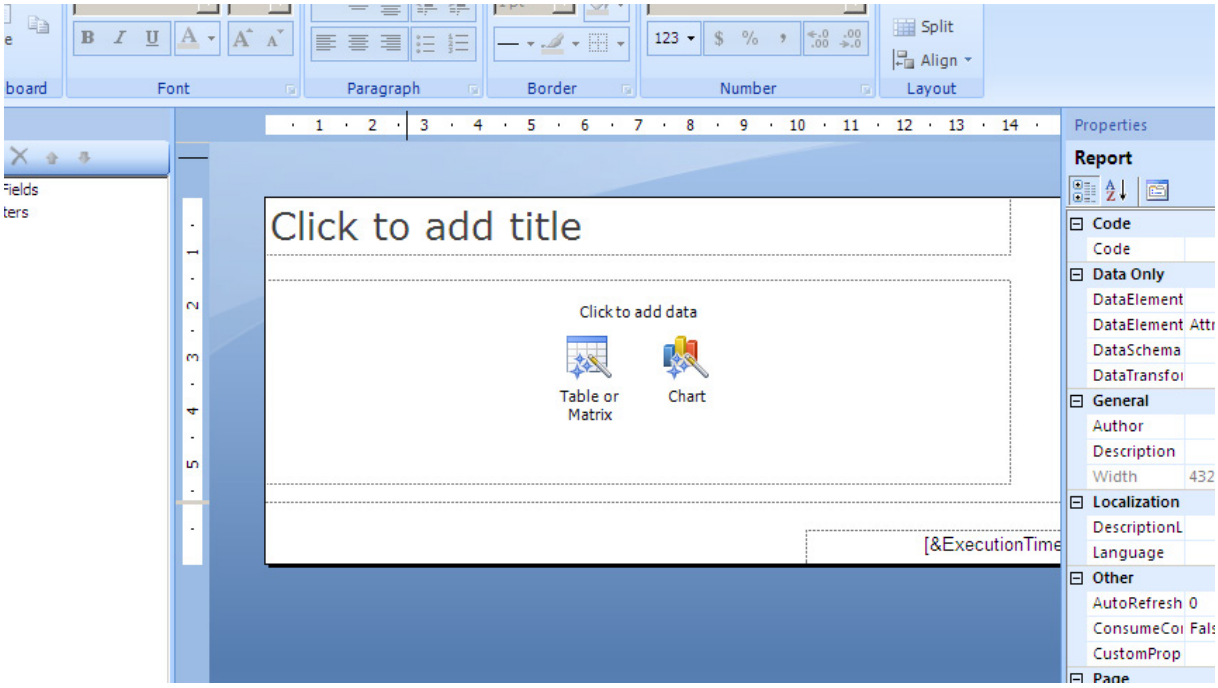

Fig. 'Click to add title' area.

The **'Choose a connection to a data source'** *window appears.*

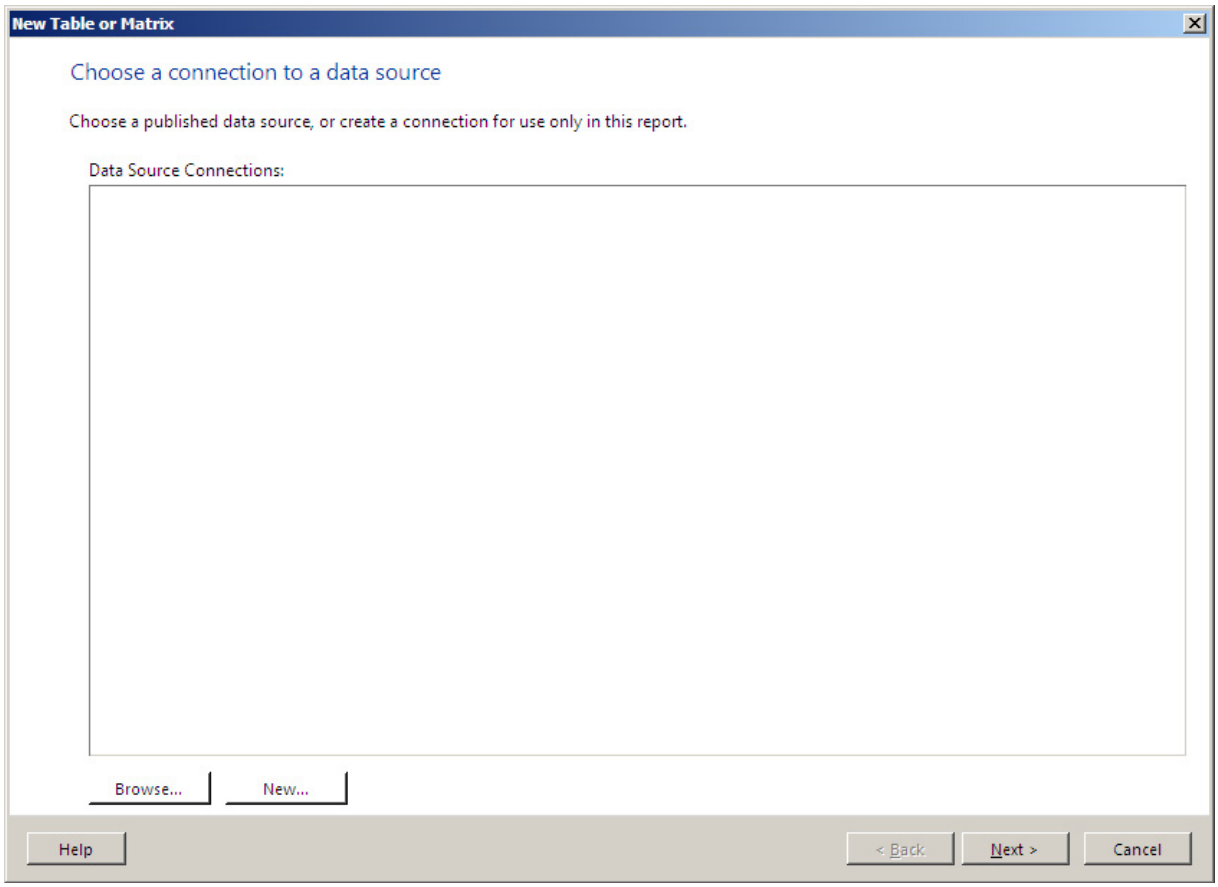

Fig. 'Choose a connection to a data source' window.

List of data sources appears, if some folder with shared source data has been defined on the report server. Otherwise such folder (data source) must be defined.

3. Click the ***New...*** *button to add a data source from the report level.*

4. Select the '**Microsoft SQL Server'** connection type, then click the ***Build*** button to define connection properties.

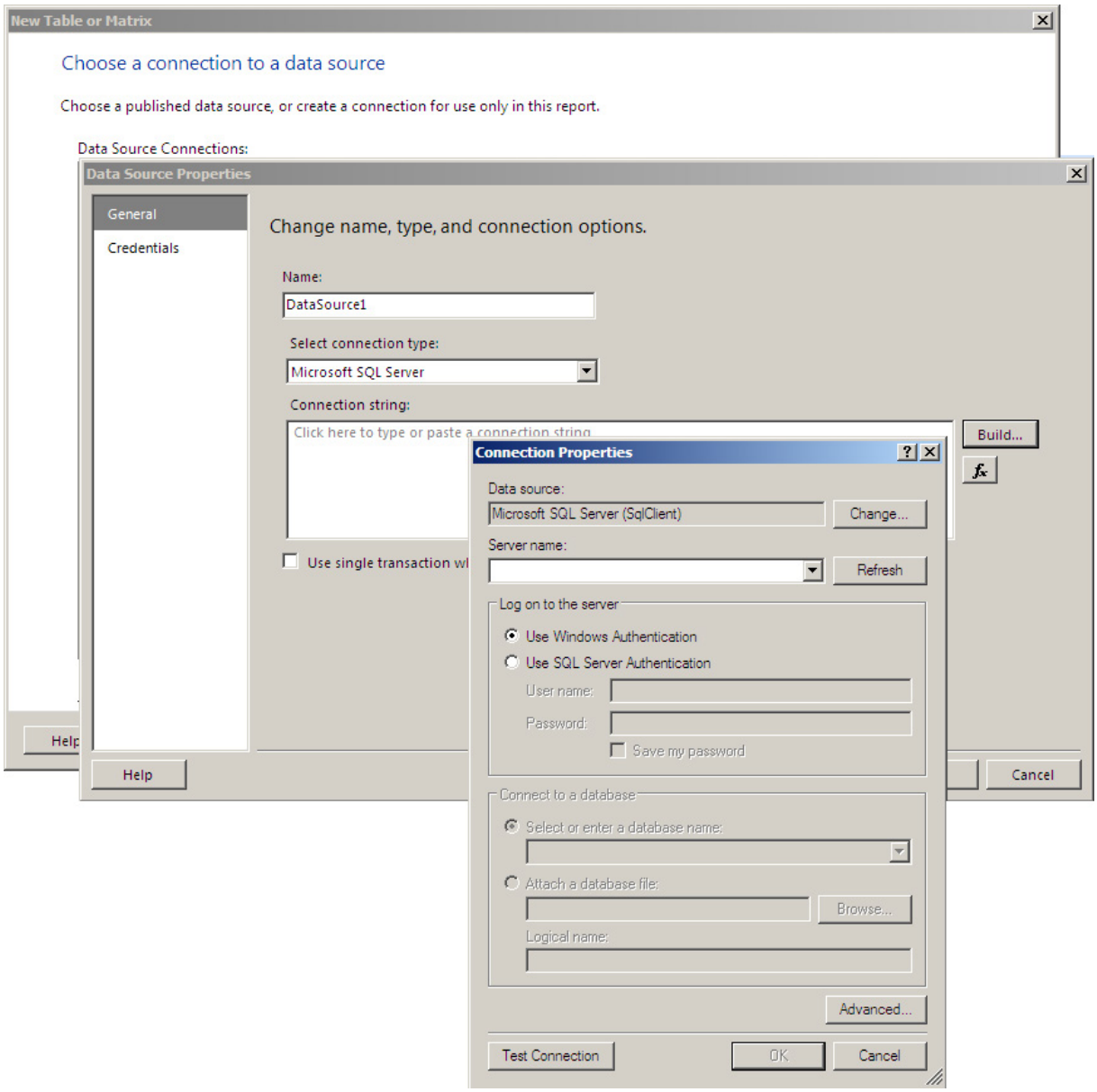

Fig. 'Connection properties' window.

5. Enter path to the AsixConnect database.

6. Click ***OK*** to close the *'Connection Properties'* window, another ***OK*** to close the *'Data Source Properties'* window, and ***Next*** to proceed with the wizard.

ATTENTION: The above procedure may be used to define data sources from the report level. However, in order to be able to freely modify reports stored on a report server, the data source should be declared on that server, too. Such sources are automatically displayed on the *'Choose a connection to a data source'* wizard screen. This may be done (for example) using the Report Manager tool, which may be invoked as follows:

Menu *Start* > *Microsoft SQL Server 2008* > *Configuration Tools* > *Reporting Services Configuration Manager* > *Report Manager URL* > *URLs*

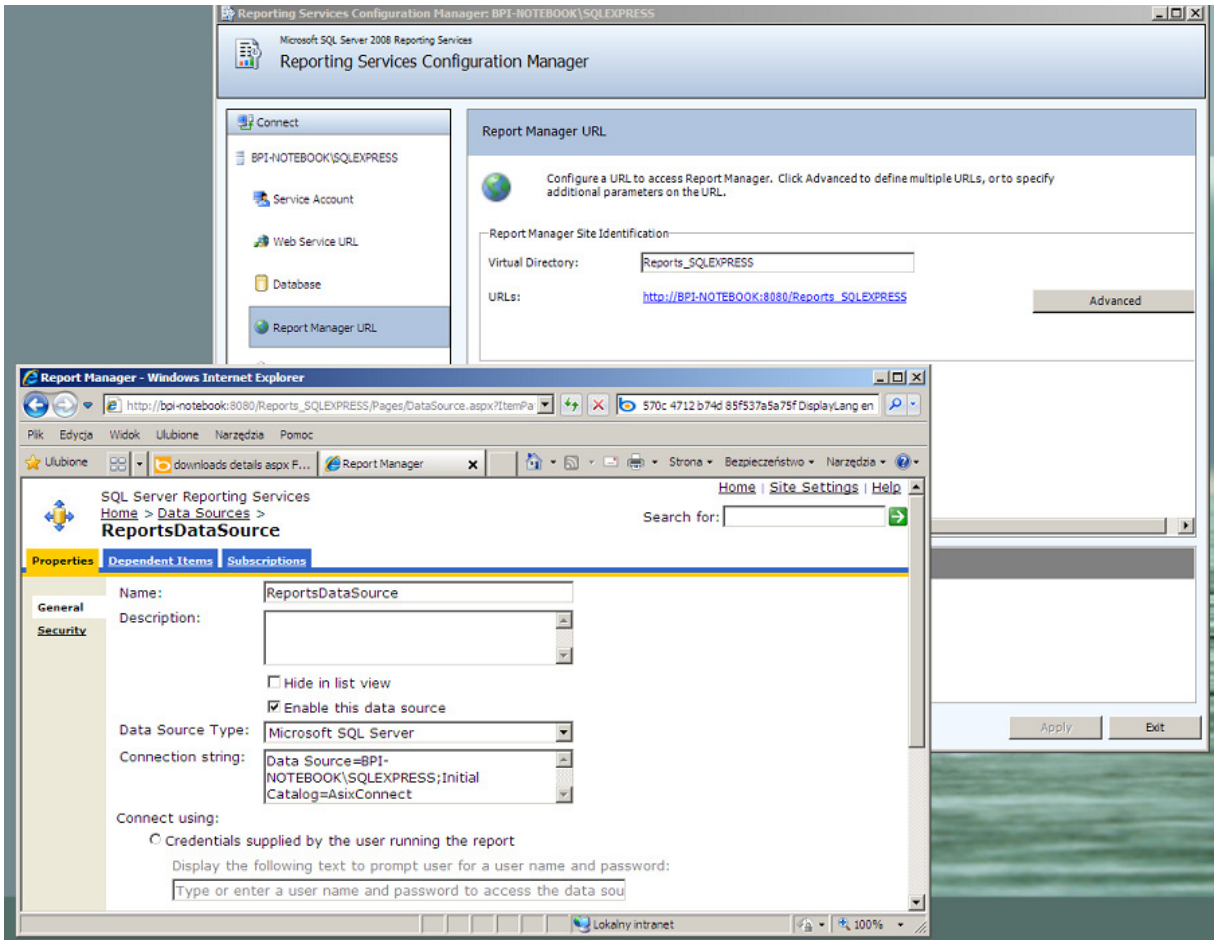

Fig. Report Manager - report data source declaration.

Once data source is declared, declare a query to retrieve the to-be-reported data. Proceed with the table wizard:

7. Once path to the AsixConnect database is declared, the table wizard will display the **'Design a query'** *window.*
8. *Provided that the query has already been created in the AsRaport* editor, copy it from the editor to the Windows Clipboard, navigate to the ***'Design a query'*** screen of the Report Builder wizard, select the ***Edit As Text*** option and paste the Clipboard contents. Click the button ! to display query results.

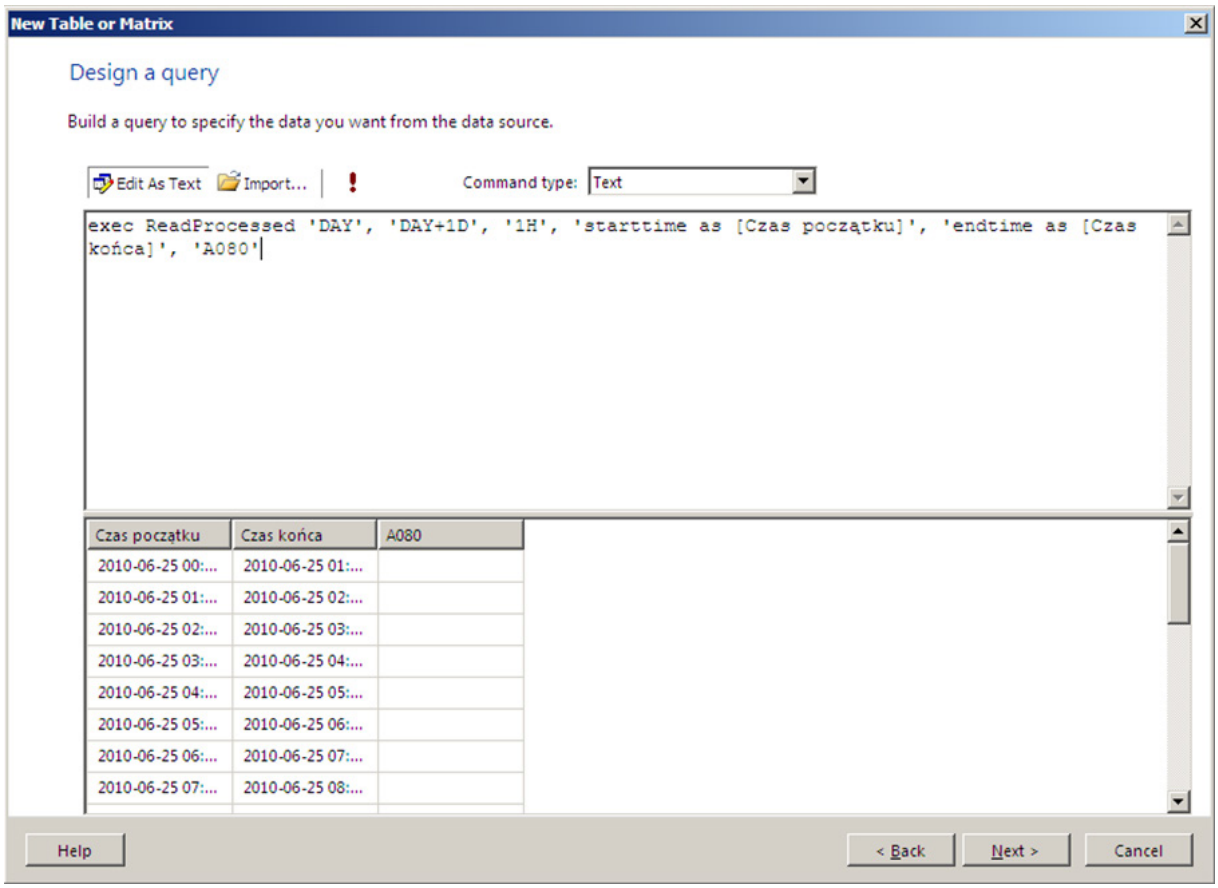

Fig. Report Bulider – table wizard – 'Design a query' window.

9. Click ***Next***.

10. Arrange the fields grouping them into rows, columns or rows/columns and click ***Next***.

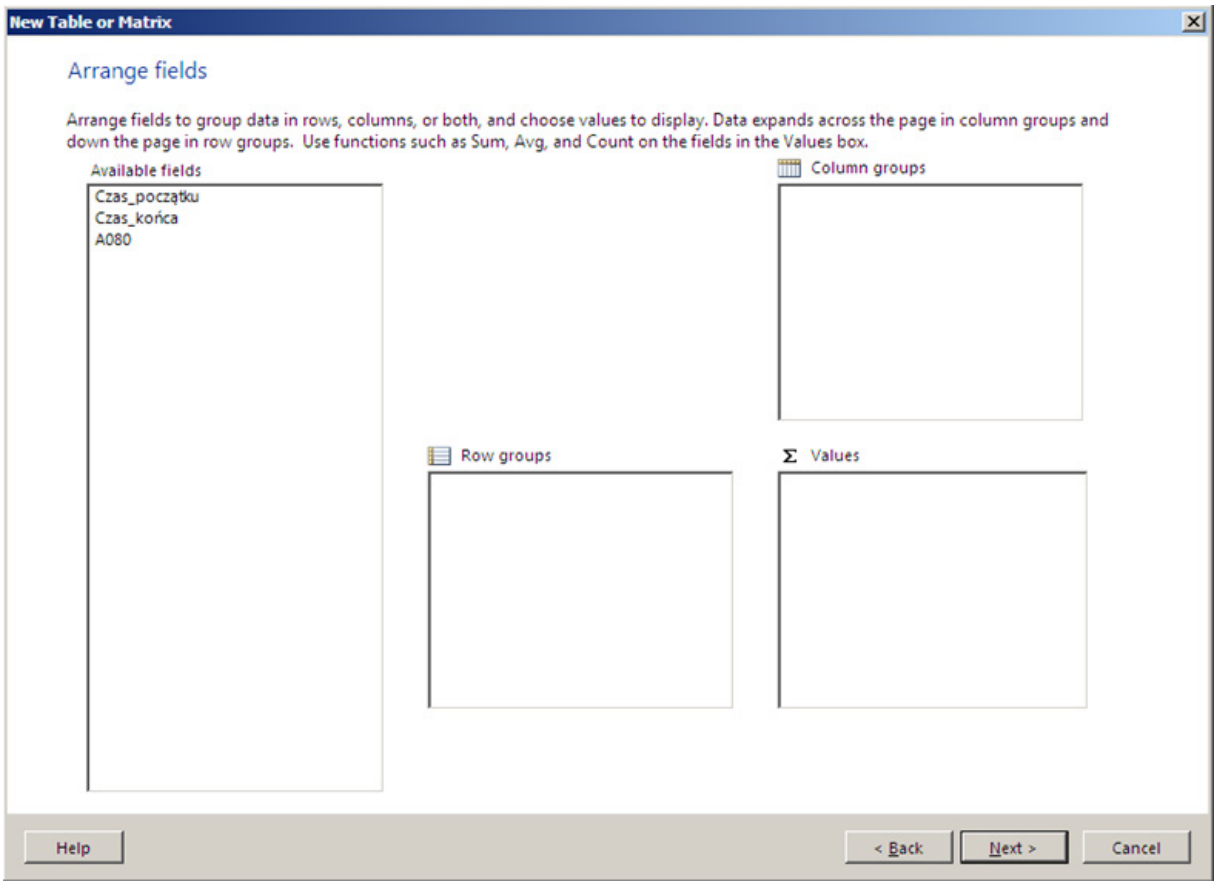

Fig. Report Bulider – table wizard – 'Arrange fields'.

11. Select a layout for the table and click ***Next***.

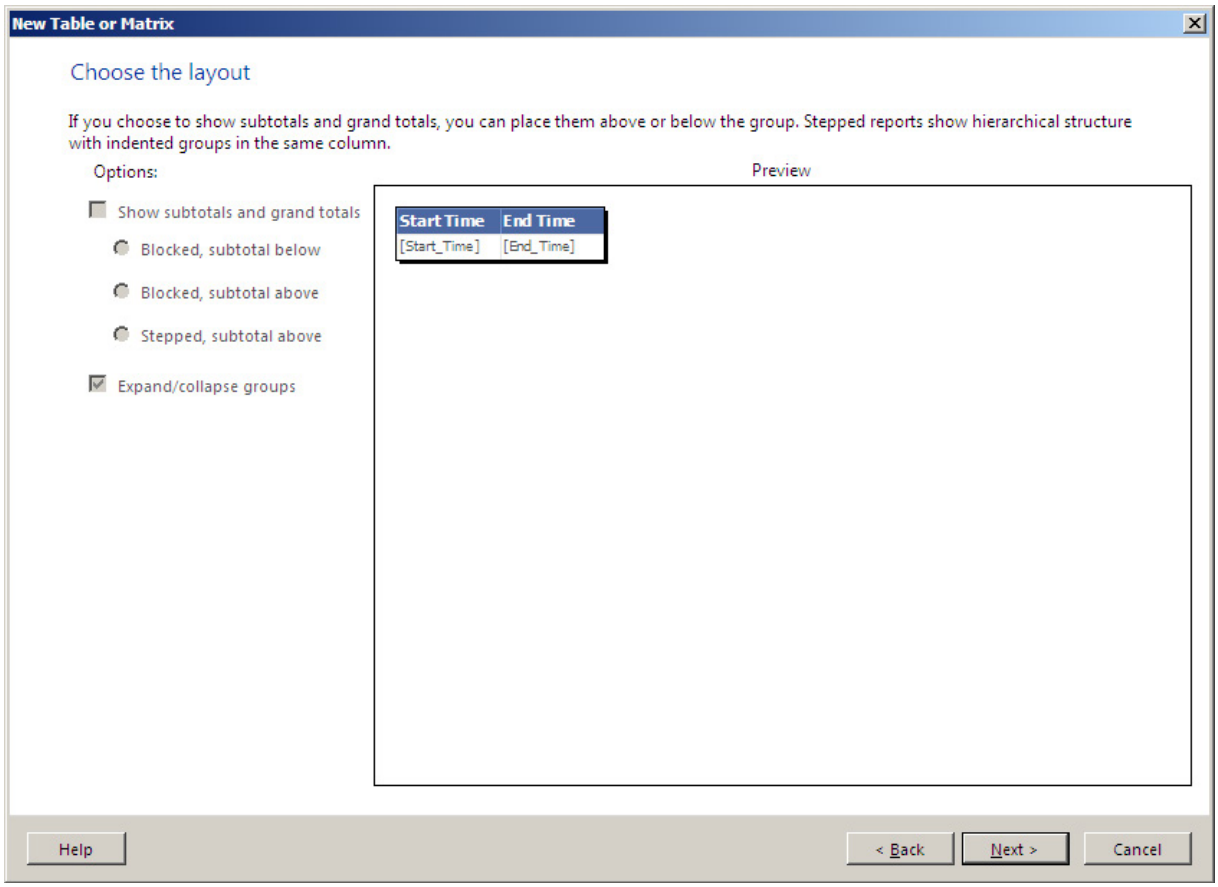

Fig. Report Bulider – table wizard – 'Choose the layout' window.

12. Select a style for the table and click ***Finish***.

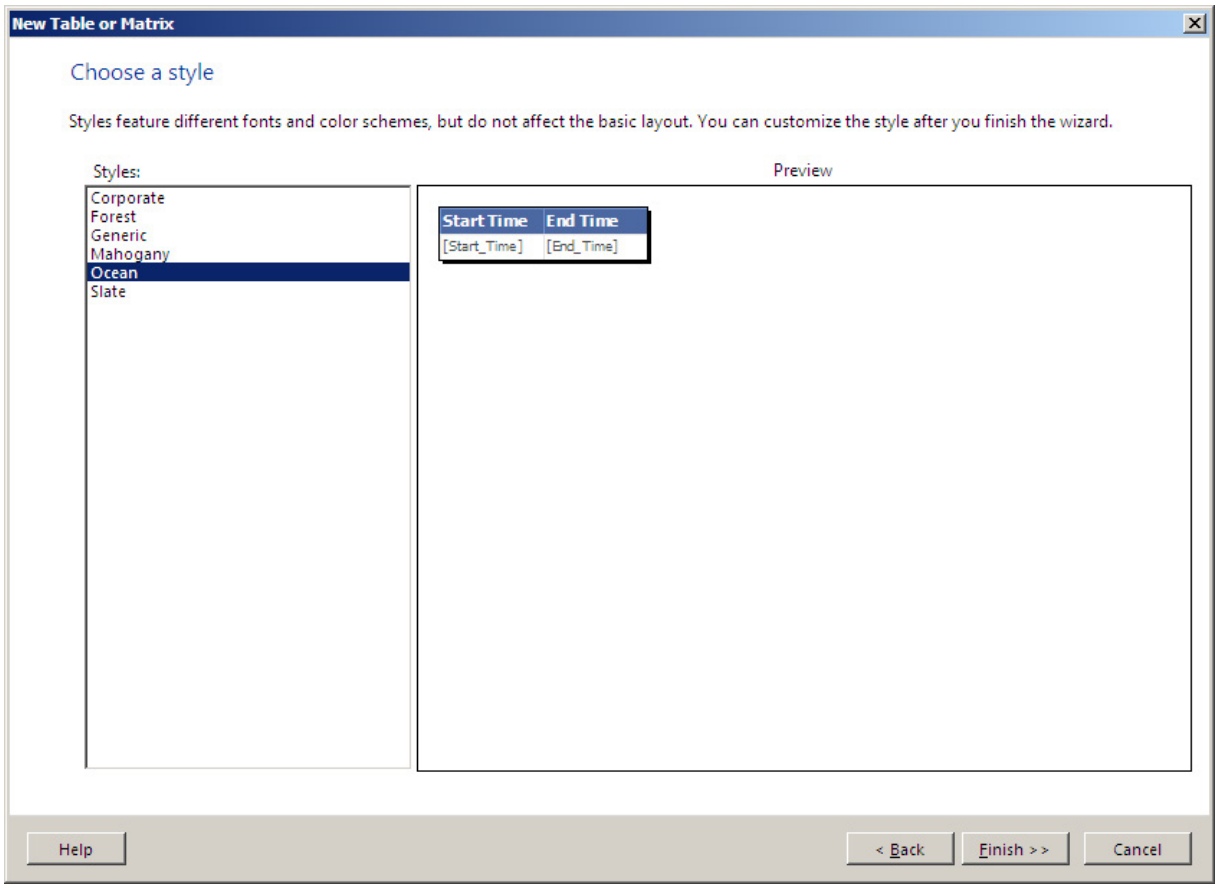

Fig. Report Bulider – table wizard – 'Choose a style' window.

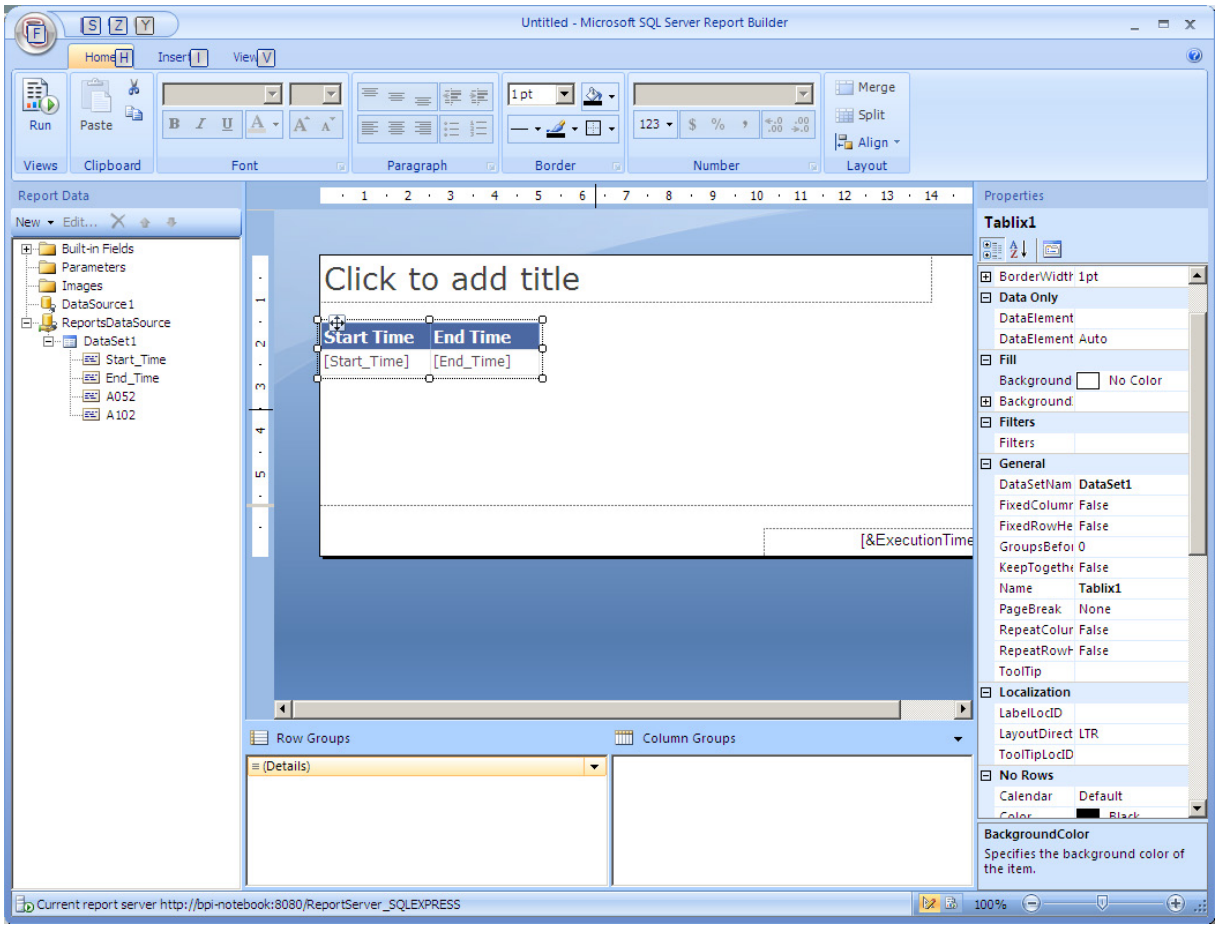

Fig. Report in 'Design' view.

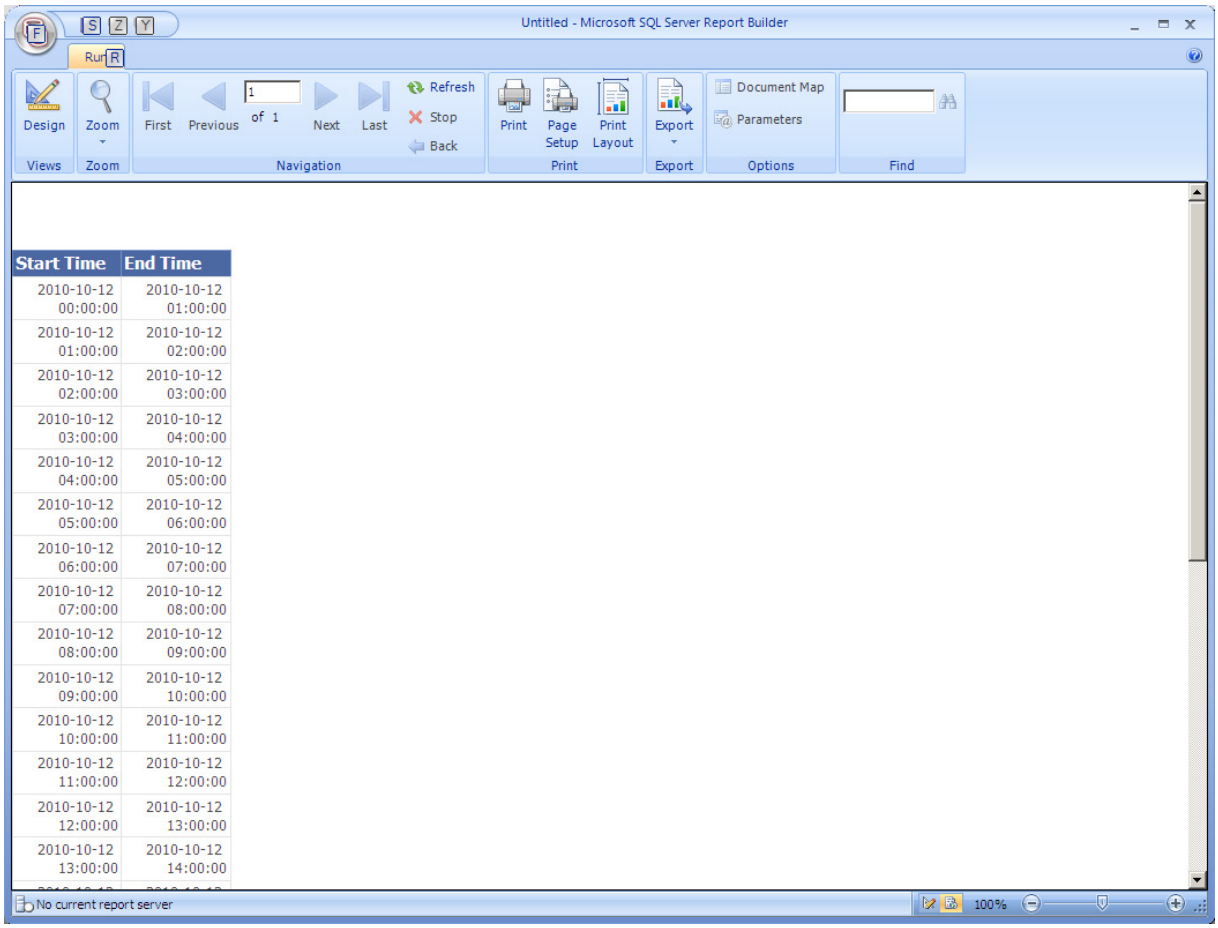

Fig. Report in 'Run' view.

4.3 View the Report

Use the ***Run*** command on the Report Builder *Home* tab to view the report.

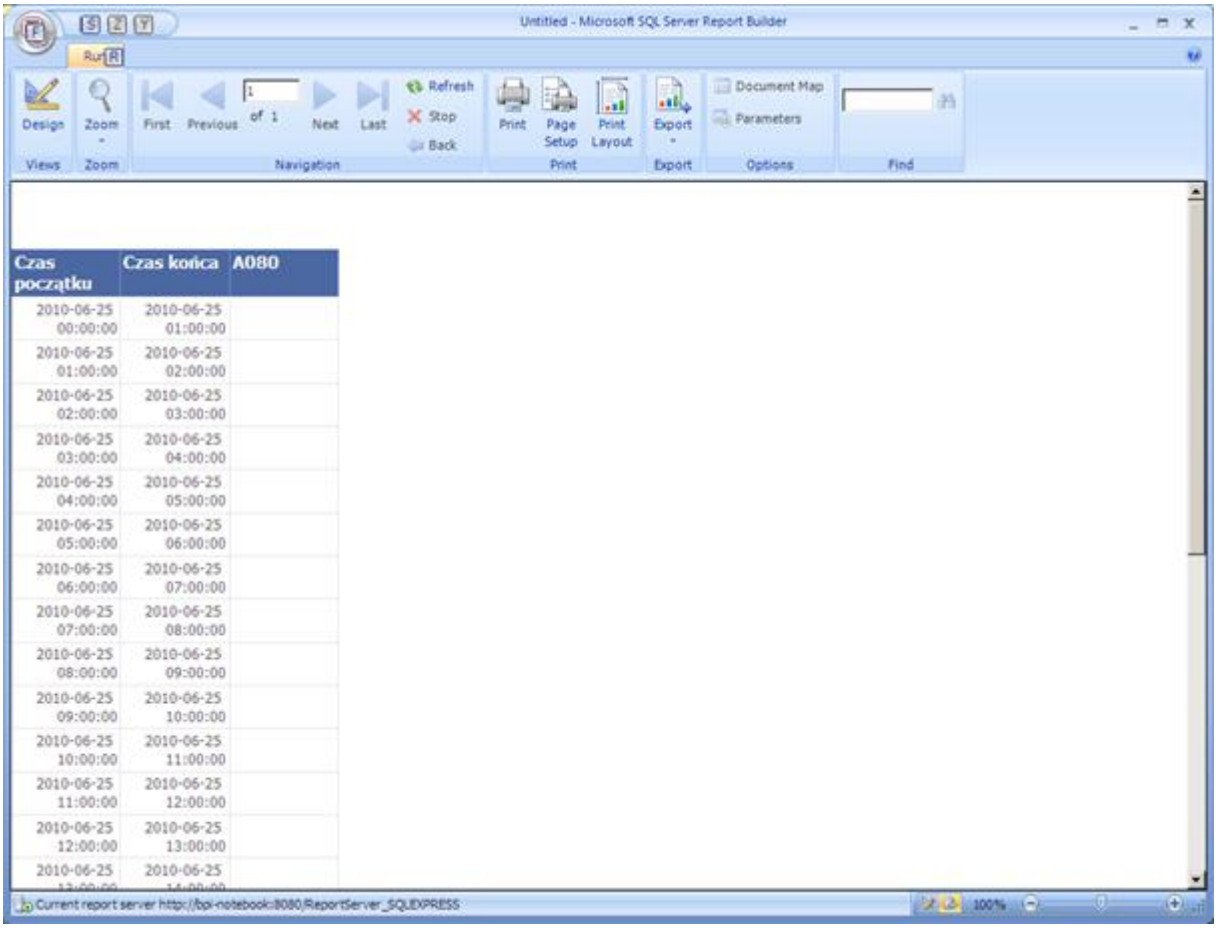

Fig. Report in 'Run' view.

4.4 Add/Edit Data Set

Data sets may be edited from the '***Report Data'*** window pane (the part of Report Builder main window).

ATTENTION: Data source must be specified.

To create a data set:

1. Right-click data source name displayed in the ***'Report Data***' window pane, then click the ***Add Dataset***… command button to open the ***'Dataset Properties***' dialog window.

If there is no data source item in the *'Report Data'* window pane select ***New*** menu and ***Data Source...*** command.

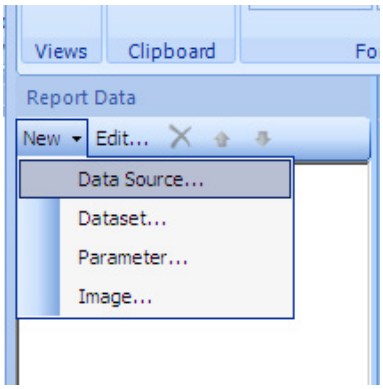

Fig. 'New' menu in the 'Report Data' window.

2. Accept the default name for the edited data set or enter another name into the ***Name*** field.
3. Select name of one of the existing and shared data sources displayed in the ***Data source*** field, or click the ***New…*** button to create a new embedded data source.
4. Depending on the data source type select the appropriate ***Query type*** option:
   - select the ***Text*** option for queries written in the data source query language
   - select the ***Table*** option to retrieve all fields of a relational database table
   - select the ***Stored Procedure*** option to run stored procedure specified by its name.
5. Enter name of the query/stored procedure/table to the ***Query*** field. Alternatively you may click the ***Query Designer…*** button to open a graphical/text query editor, or the ***Import***… button to import the query from some existing report.
6. Enter maximum number of seconds the report server is to wait for a replay from the database into the ***Time out*** field. Default value 0 means that there will be no limit at all.
7. Click ***OK***.

The data set and its collection of fields is displayed in the *'Report Data'* window pane in the data source node of the Report Builder main window.

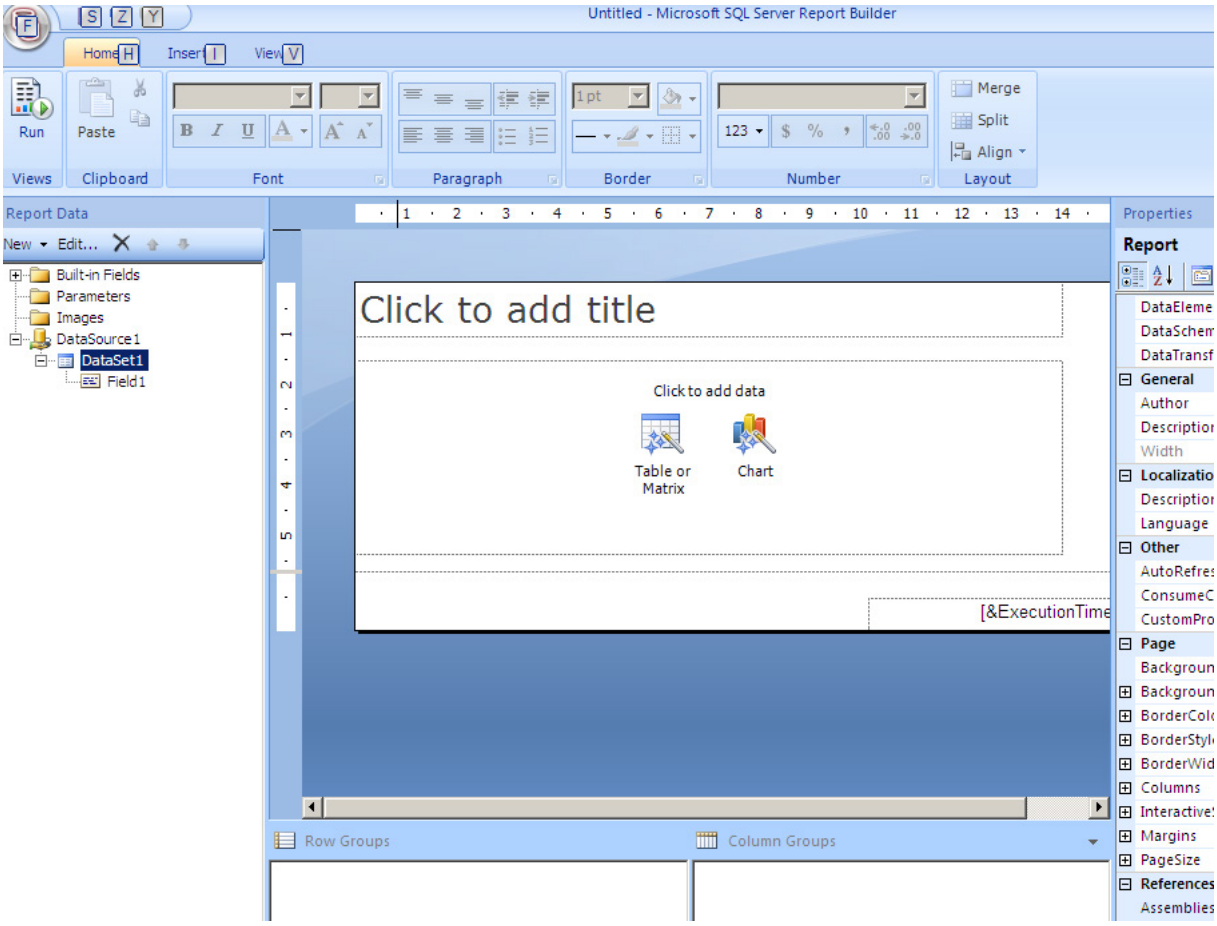

Fig. The data set and its collection of fields in the 'Report Data' window pane.

To edit a dataset:

Right-click the data source name displayed in the *'Report Data'* window pane, then right-click the data set name and click the ***Dataset Properties*** button to open dataset properties window in the query properties view.

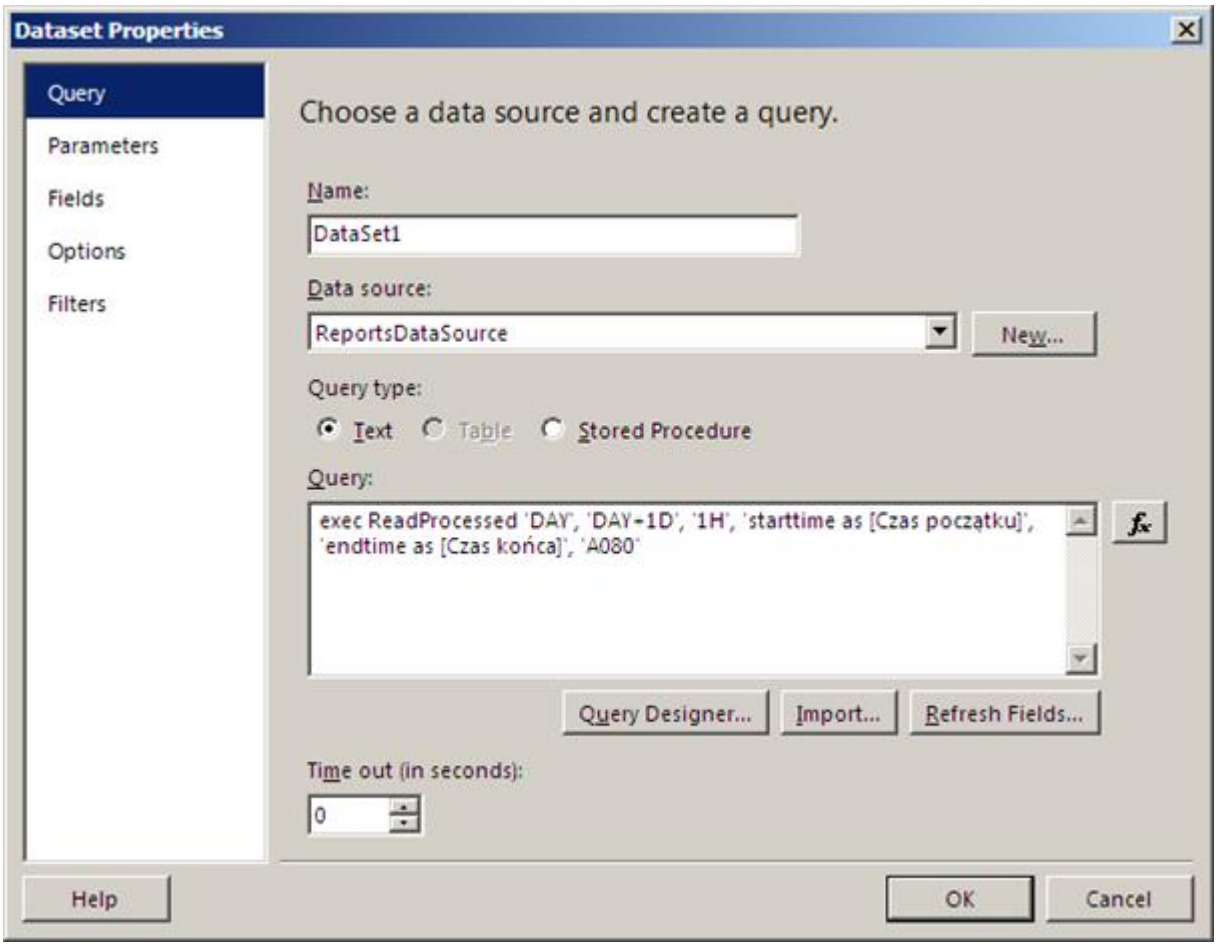

Fig. 'Dataset Properties' window.

Select another category in the left window pane to switch to another dataset property.

4.5 Embedded vs. Shared Sources of Data

Report Builder 2.0 can link a report to a data source shared on a report server or embed a data source into a report; in the latter case the data will be used exclusively by the report.

To be able to link a report to a data source shared on a report server one has to have access to the server and to know data source location within the server. To embed data source one has to specify connection and to know what privileges to specify so that the data needed within the report might be retrieved.

Once a shared data source is selected on a report server, the server becomes the current report server.

To create a link to a shared data source (shared connection):

1. Click the ***New…*** menu bar option in the ***'Report Data'*** (left) window pane of the Report Builder main window, then click the ***Data Source…*** option to open the ***'Data Source Properties'*** dialog window.

2. Accept the default name for the edited data source or enter another name into the ***Name*** field.

3. Set the ***Use a shared connection or report model*** option to display list of shared data sources and report models used in the report. If the list is empty, click the ***Browse…*** button and navigate to the report server folder, in which the shared data sources reside.

4. Highlight the shared data source and click ***OK***.

Data source name will appear in the ***'Report Data'*** window pane.

To embed a data source:

1. Click the ***New…*** menu bar option in the ***'Report Data'*** (left) window pane of the Report Builder main window, then click the ***Data Source…*** option to open the ***'Data Source Properties'*** dialog window.

2. Accept the default name for the edited data source or enter another name into the ***Name*** field.

3. Make sure the ***Use a connection embedded In my report*** option is selected.

4. Drop down the ***Select connection type*** list and select data source type e.g. 'Microsoft SQL Server' or 'OLE DB'.

5. Using one of the following methods specify the connection string:

   - enter the string directly into the ***Connection string*** text box
   - click the ***fx*** formula button to display the '***Expression***' dialog window, enter expression into the ***Set expression for***: box, click ***OK***.

6. Click the ***Build...*** button to open the ***'Connection Properties'*** dialog window with properties of the above specified connection.

7. Depending on the selected data source type fill up appropriate fields of the ***'Connection Properties'*** dialog window (data source type/name, credentials used to connect etc.). Click the ***Test Connection*** button to make sure that the data source is available and that the supplied credentials are correct.

8. Click the ***Advanced*** butoon - the ***'Connection Properties'*** window will appear.

9. Specify credentials to be used to connect to the data source. Data owner is responsible for selecting acceptable credentials. Sometimes the owner decides to share data source on a report server and sets (in the server properties) up credentials available for other users. Ask data owner for the credentials.

10. Click ***OK***.

Data source name will be displayed in the ***'Report Data'*** window pane.

4.6 Save the Report

To save the report:

1. Click the icon in the top left corner of the Report Builder main window and select the ***Save As*** command.

2. Specify URL address of the target server report or a local computer target folder. The report will be saved in a *.RDL file.

3. Click the ***Save*** button.

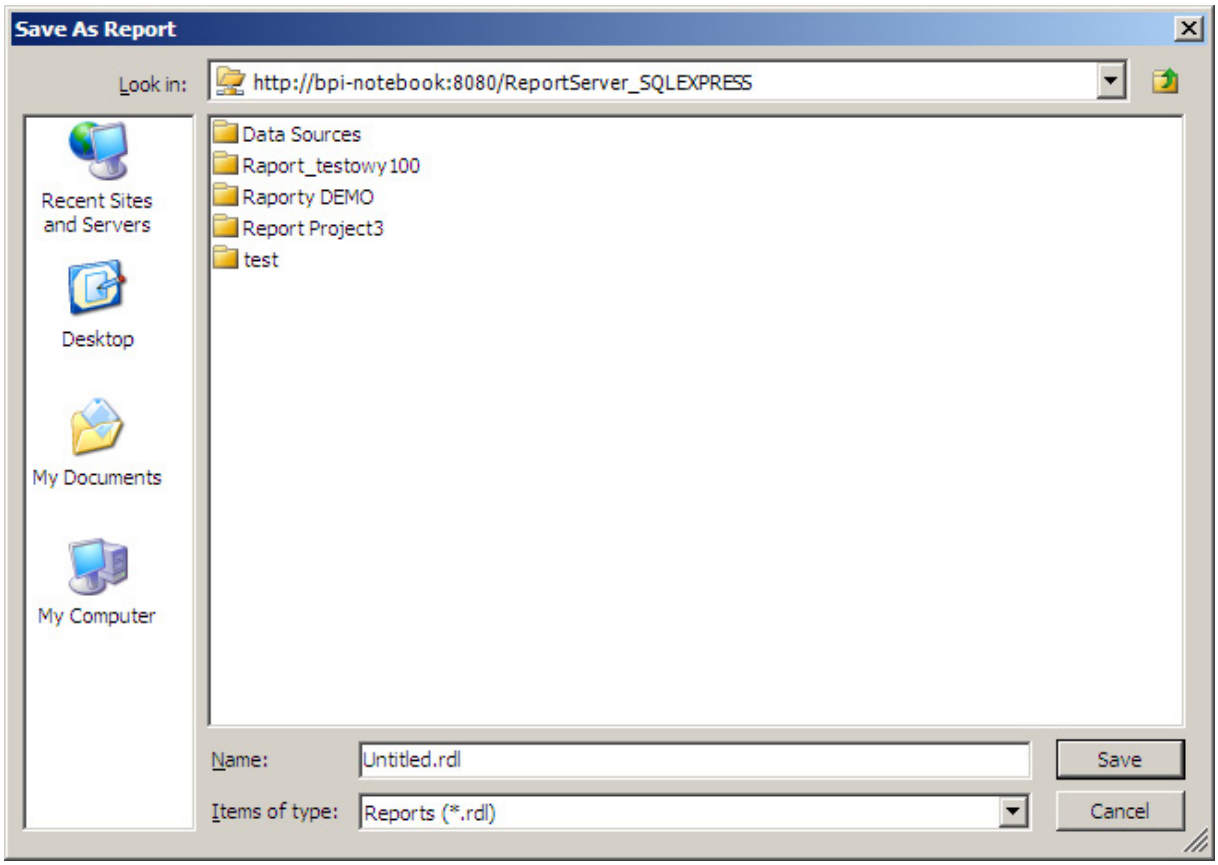

Fig. 'Save As Report' window.

5 Designing Reports in Report Designer

Only basic steps of the procedure to define simple reports utilizing process data sources from Asix system applications have been described in this section. The “Factory” demo application delivered with the Asix package has been used as a sample application. Services from the SQL Server 2008 Service Pack 1 version Express have been used. For detailed instructions see documentation of the Report Designer program.

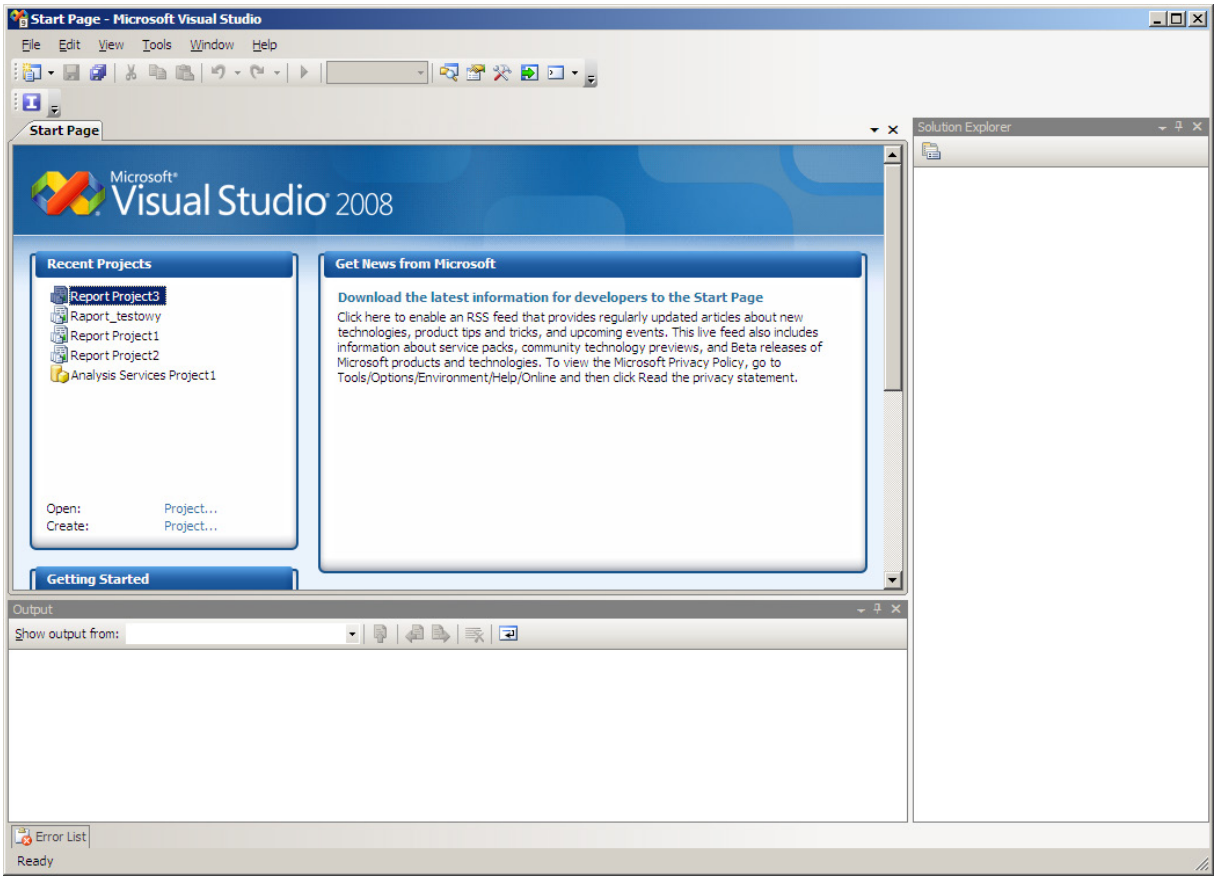

Fig. The main Report Designer window.

1. Run *Start* > *Programs* > *Microsoft SQL Server 2008* > **SQL Server Business Intelligence Development Studio**
2. Select the ***Create: Project…*** option in the ***'Recent Projects'*** window pane.
3. Enter basic report parameters: its name and location. Click ***OK***.

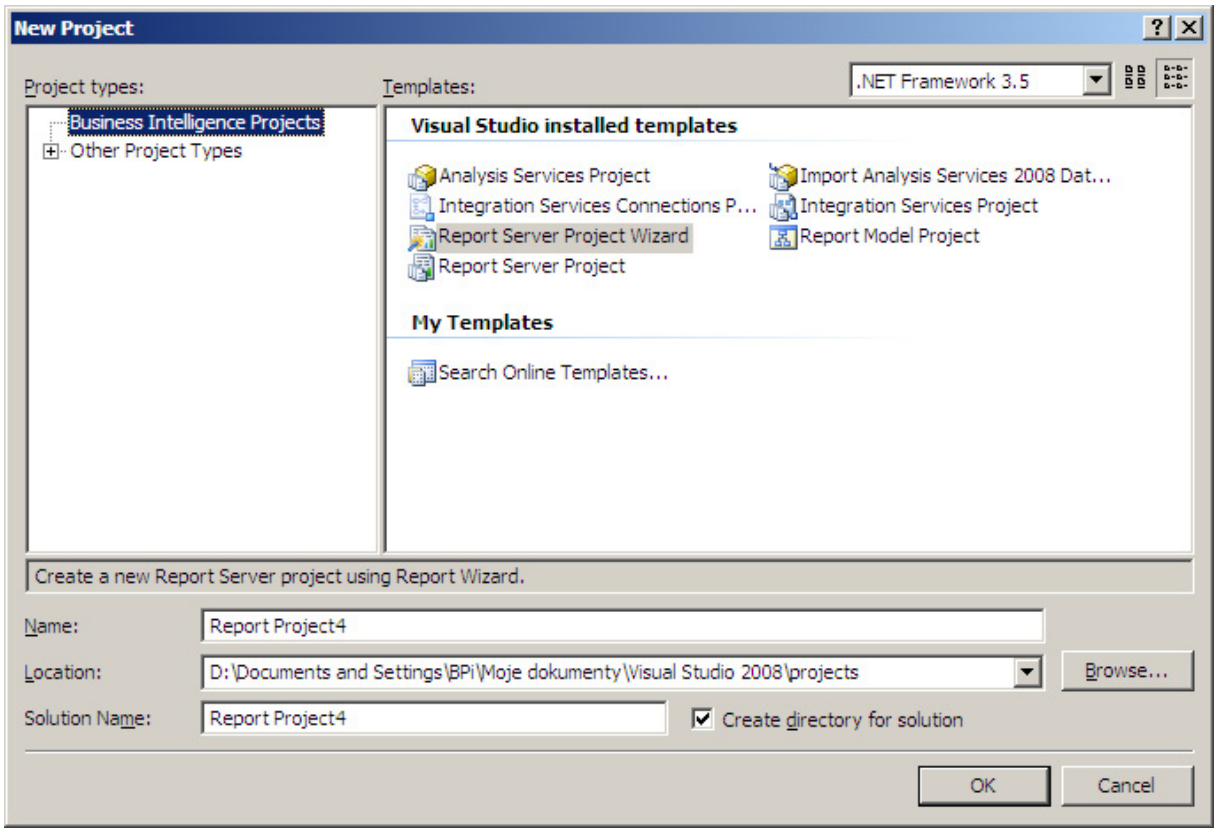

Fig. 'New Project' window.

4. Report Wizard welcome screen appears. Click ***Next***.

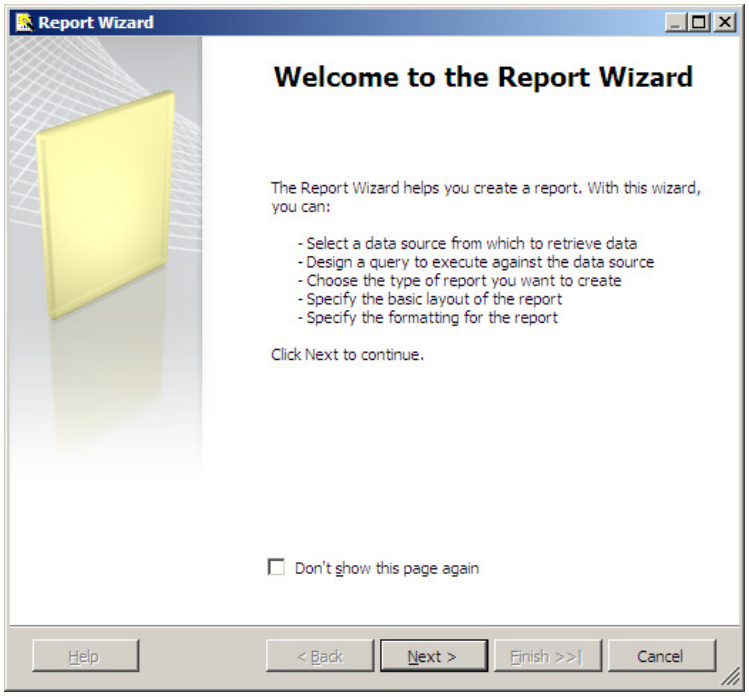

Fig. Report Wizard - greeting window.

5. On the ***'Select the Data Source'*** screen specify data source: name of the SQL server serving the AsixConnect database and the AsixConnect database - use ***Edit...*** button to open ***'Connection Properties'*** window. Click ***Next***.

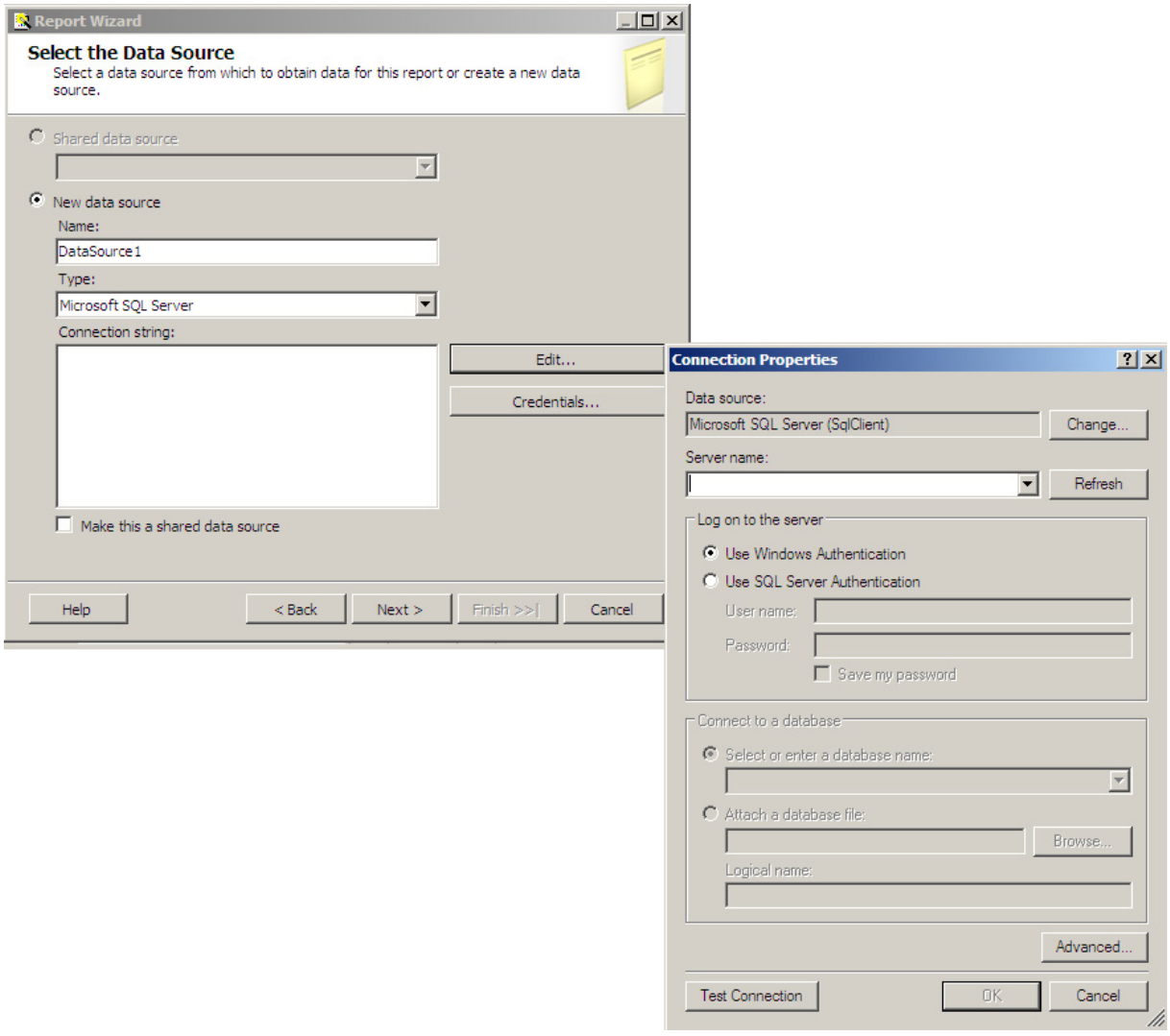

Fig. Report wizard - 'Connection Properties' window.

6. The ***'Design the Query'*** screen appears. Provided that the query has already been created in the AsRaport editor, copy it from the editor to the Windows Clipboard, navigate to the ***'Design the query'*** screen, and paste the Clipboard contents into the ***Query string:*** field.

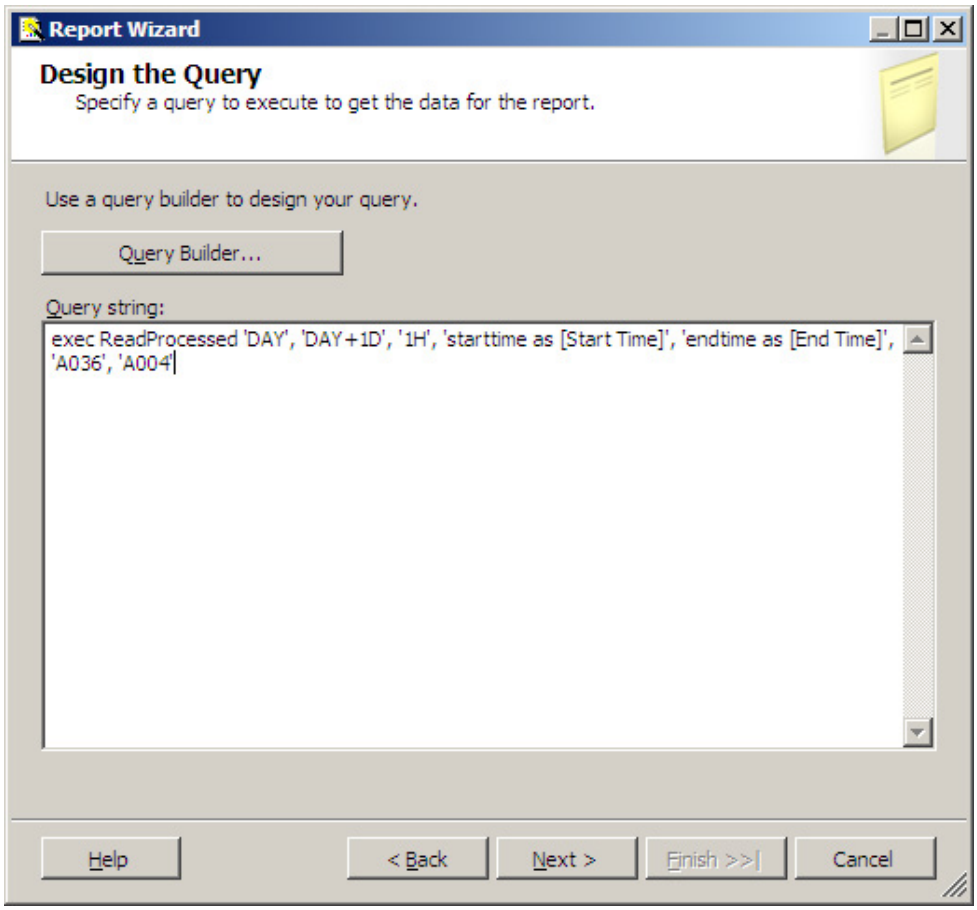

Fig. Report wizard - 'Design the Query'.

7. Select report type: 'Tabular' or 'Matrix'. Click ***Next***.

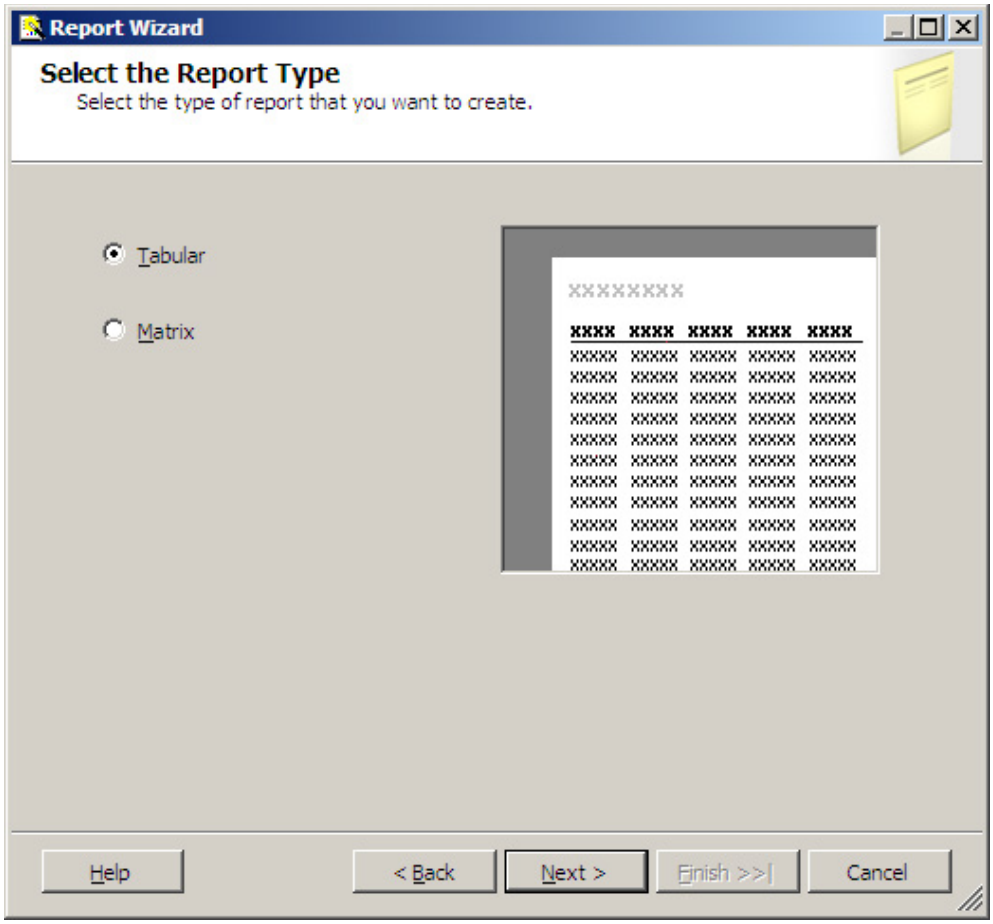

Fig. Report wizard – 'Select the Report Type' window.

8. Design a layout of the data within the report (a table was selected in this example). Click ***Next***.

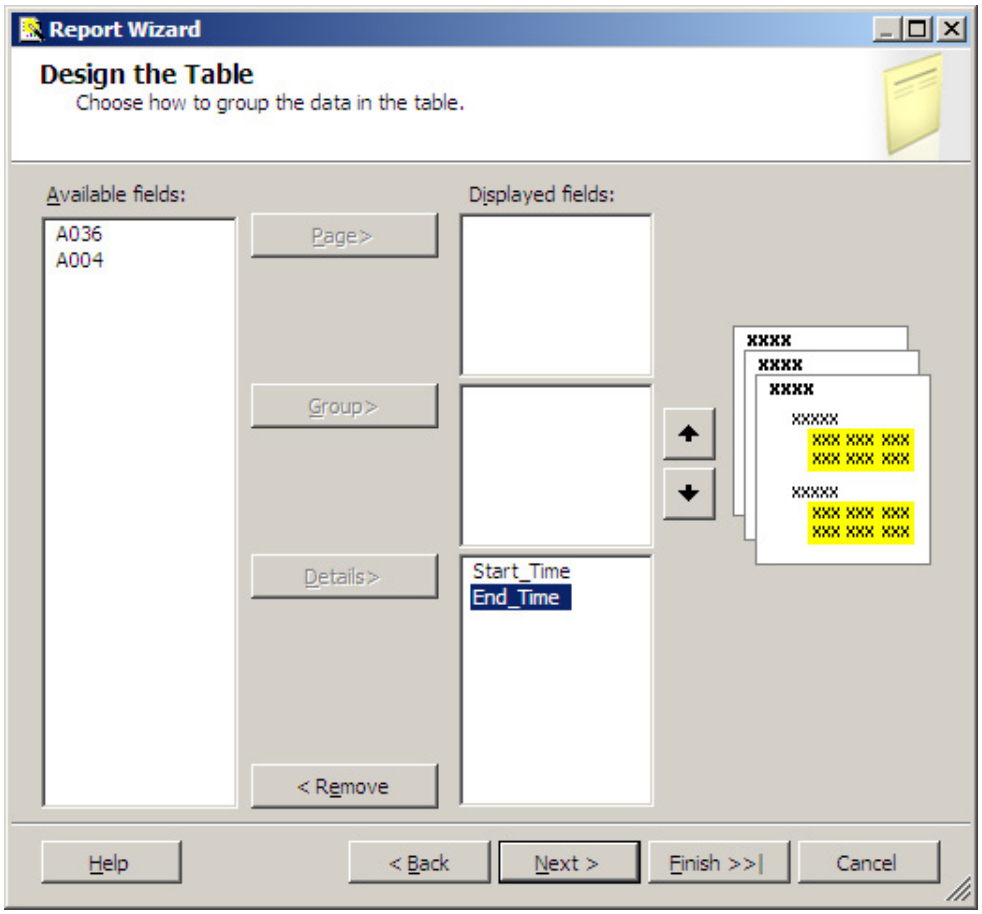

Fig. Report wizard – 'Design the Table'.

9. Select table style. Click ***Next***.

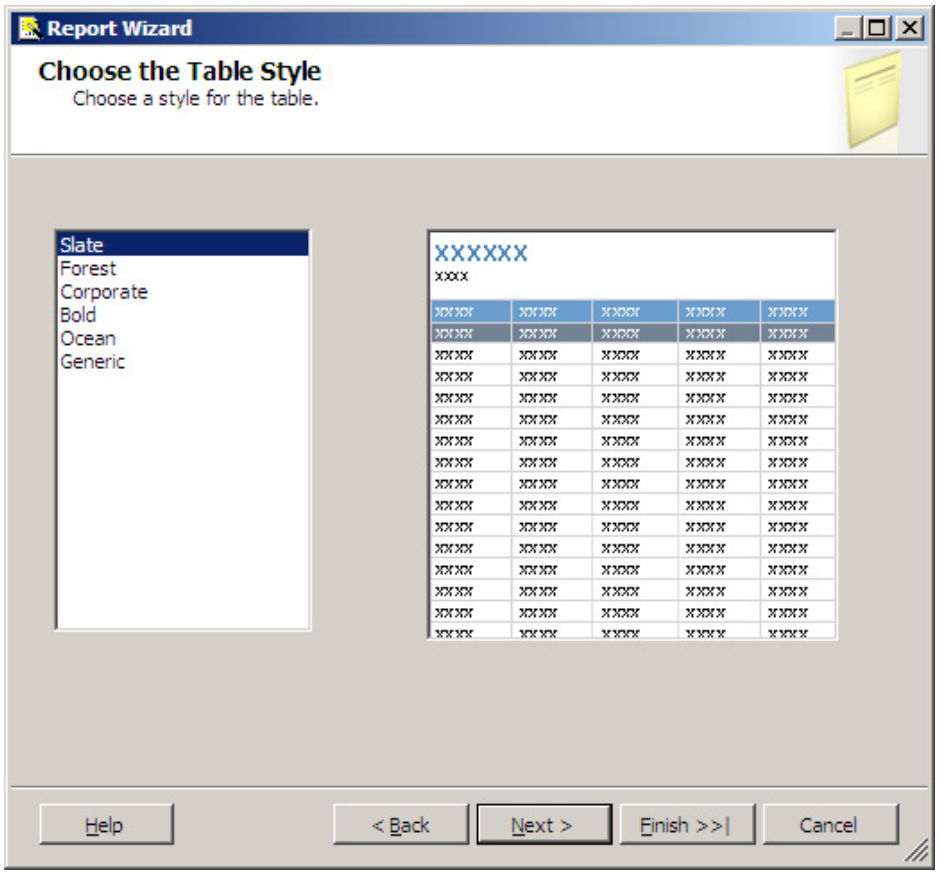

Fig. Report wizard – 'Choose the Table Style'.

10. Specify report server and its folder, to which the reports will be transferred (deployed). Click ***Next***.

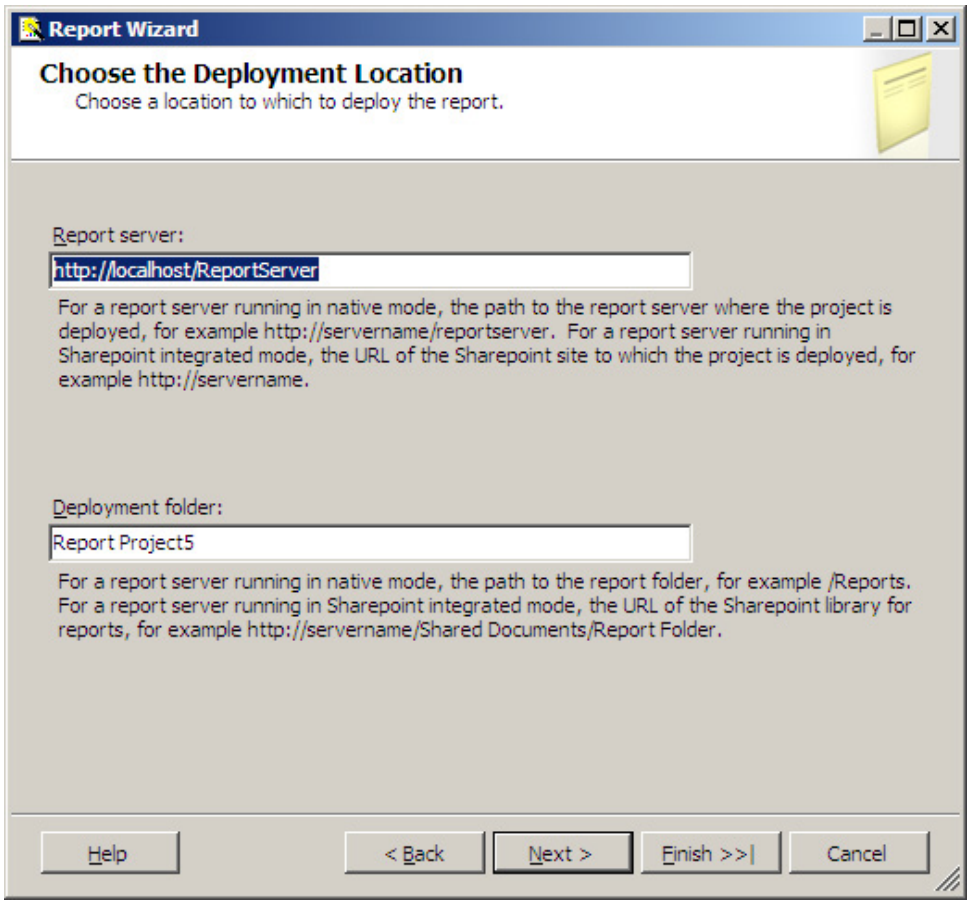

Fig. Report wizard – 'Choose the Deployment Location' window.

11. Specify report name. Click ***Finish***. The report will be displayed in the project view.

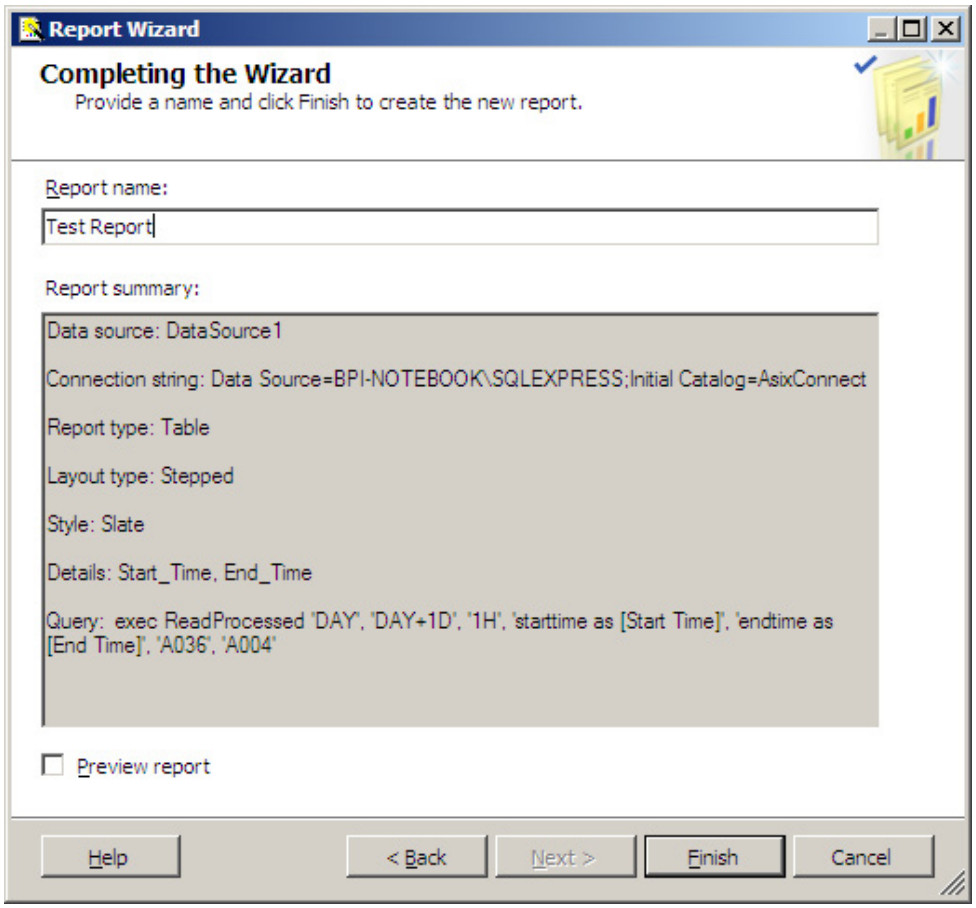

Fig. Report wizard – 'Completing the Wizard' window.

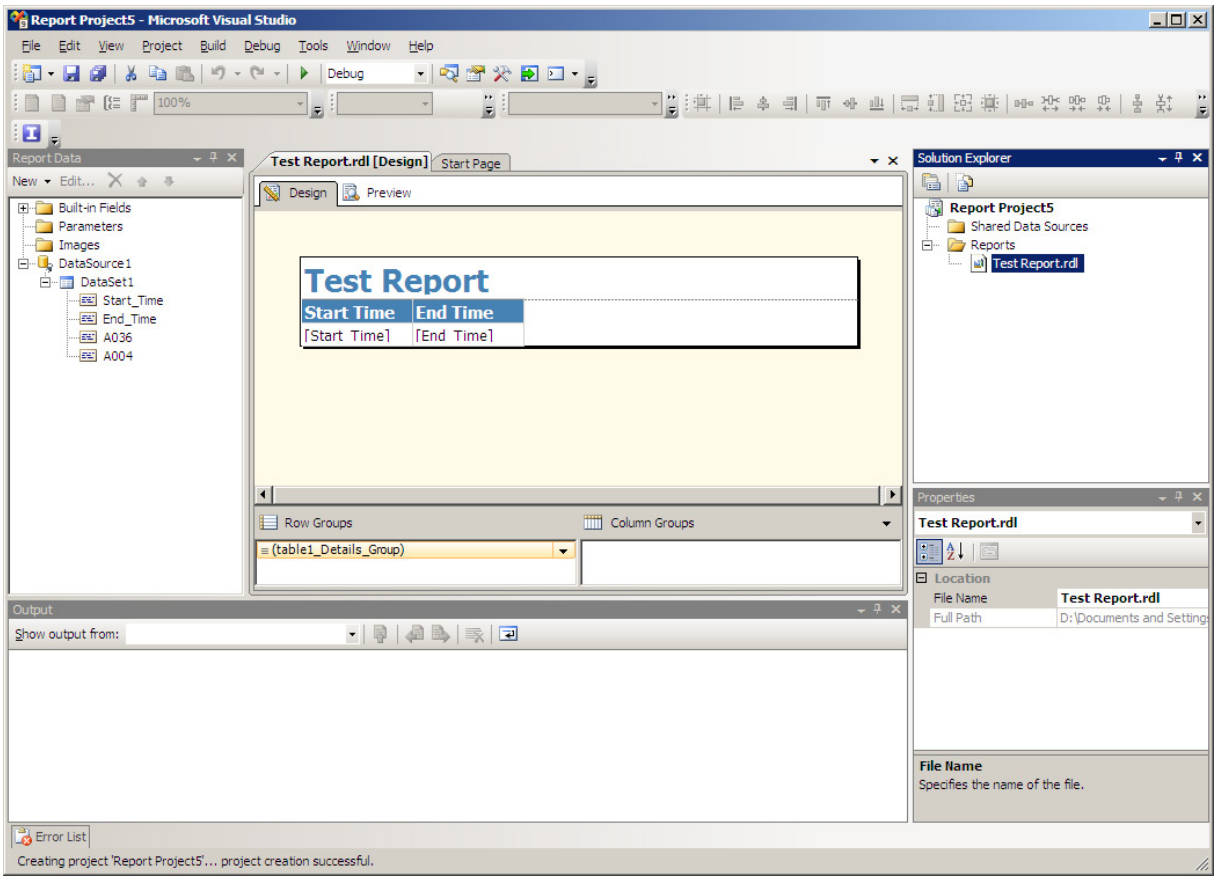

Fig. Final report in design view.

12. Click the ***Preview*** button to view the report.

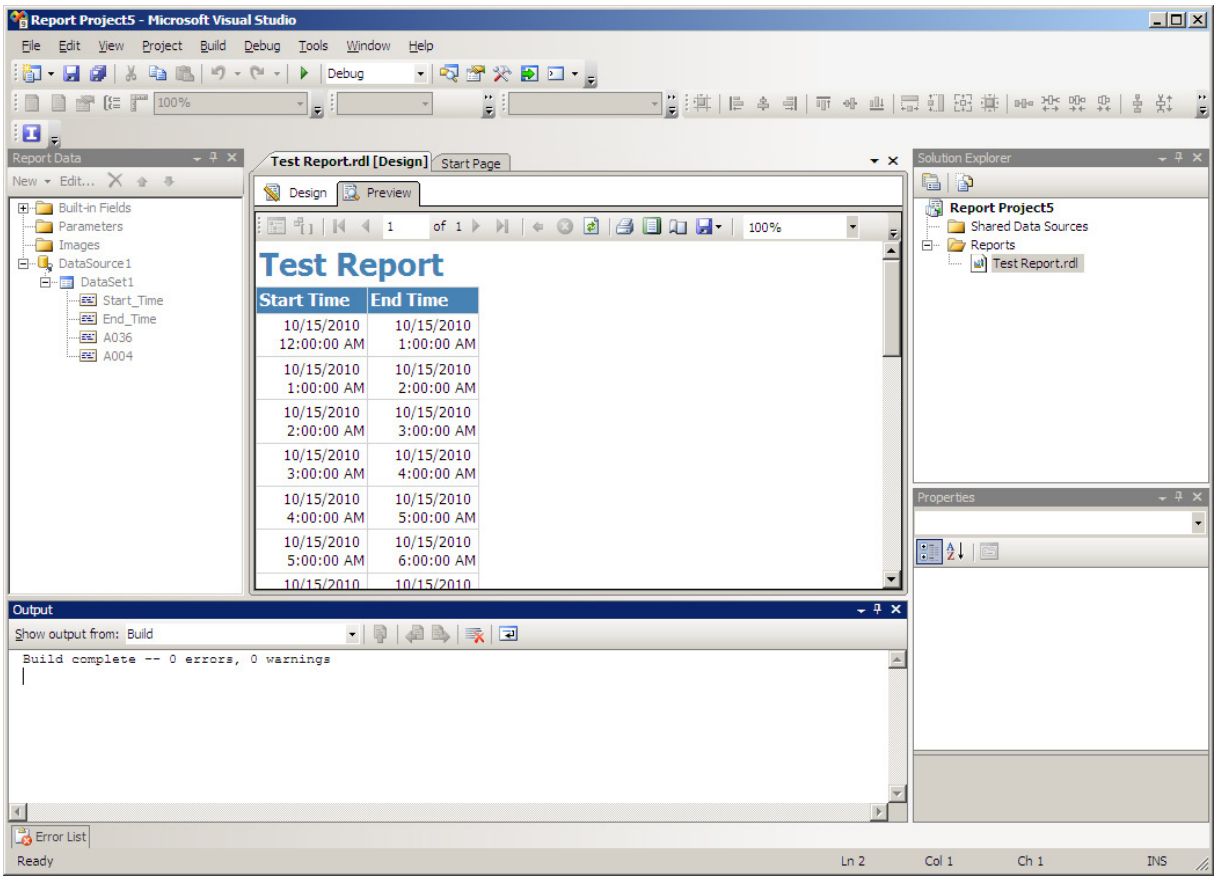

Fig. Final report in preview.

13. Select the ***Deploy <report_name>*** option of the ***Build*** menu to store the report on the report server.

6 Access to Reports Stored on Report Server

Reports shared on a report server may be viewed in any Internet browser. To view a report, specify URL address of the server. In case of the MS SQL Server 2008 Reporting Services environment the address for report server is:

http://<*computer_name*>/ReportServer_<*SQL_server_instance_name*>

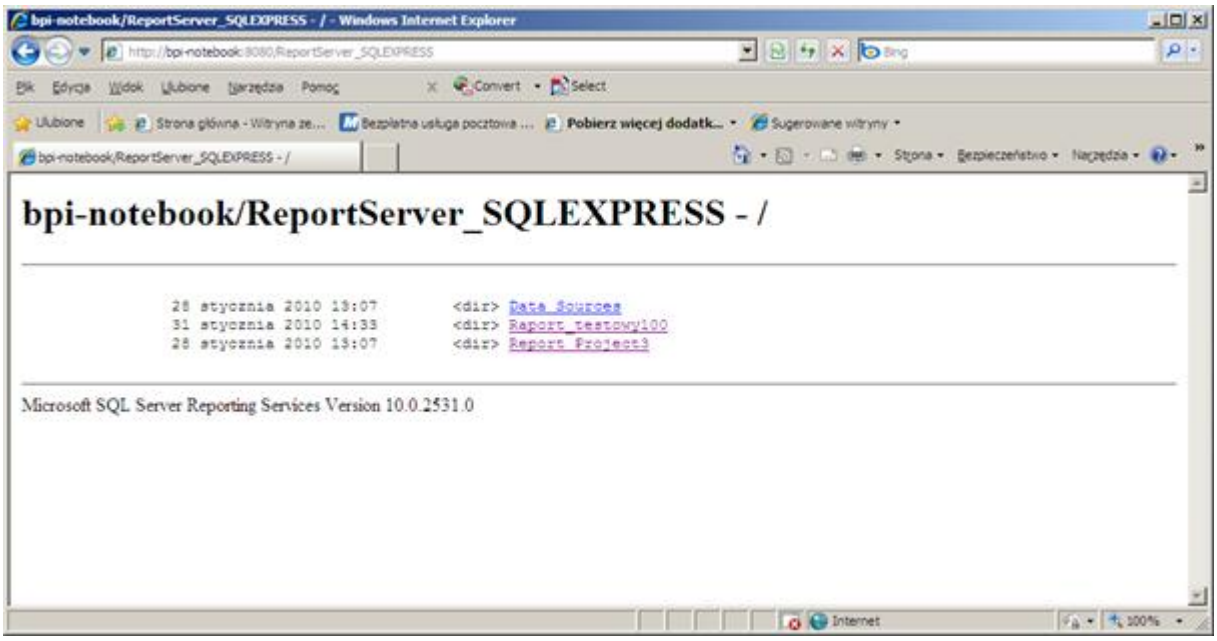

Fig. Access to reports stored on report server via Internet browser.

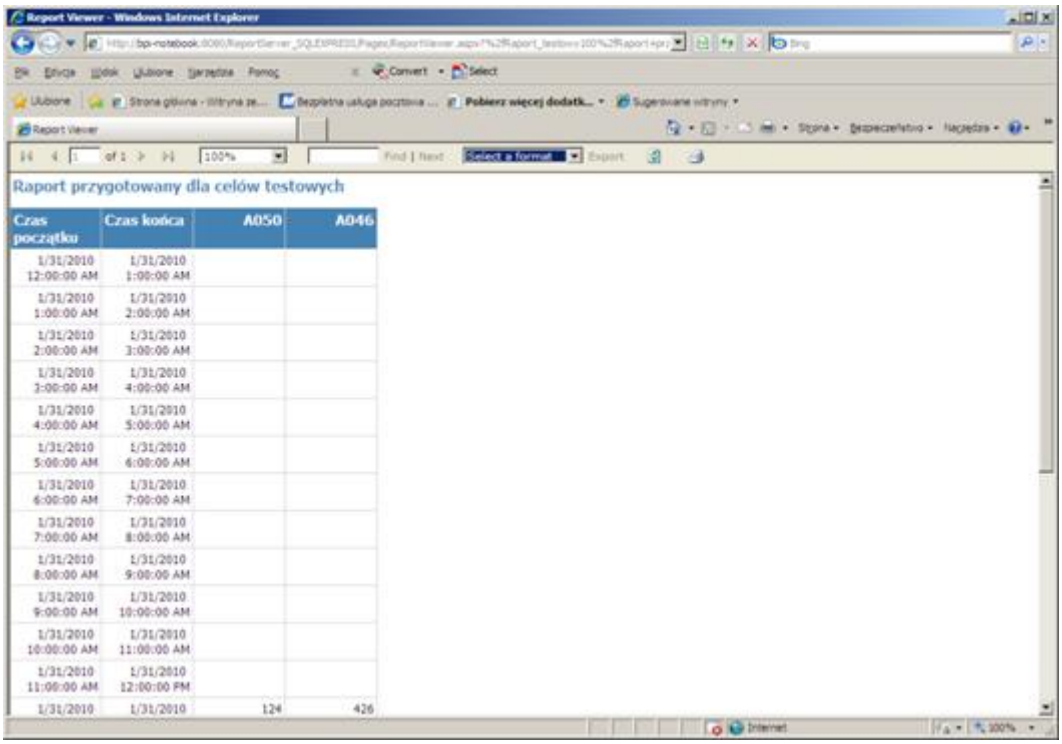

Fig. Final report in Internet browser.

Reports accessed via Internet browsers may be printed and/or exported to PDF, Excel and/or Word formats.

7 Viewing Reports Directly from Asix Applications

Asix – classic version

Reports created within the MS SQL Reporting Services environment may be directly viewed from the Asix system application level using the **AsRapView** report viewer. The viewer may be started by the ASRAPORT.

Parameter defined with the use of action editor:

Report name - name of the report created with the use of Reporting Services (the report definition stored in *.RDL file must be published on the report server of Reporting Services).

Optional configuration file - the parameter enables the declaration of XML configuration file located on other report server or in other directory than the ones declared in the configuration file of the current Asix system application.

By default, information on report server and directory are retrieved from the configuration file of the current Asix system application.
The configuration file enables declaration of 1 server and 1 directory on that server. Server and directory are declared using the configurator of the reporting system AsReport, available in Architect:

Architect > Application > Configure Reporting system AsReport... > Reports Browser tab

To display a report from another source (server, catalogue), you need to create a new configuration file for the Asix application with a different server or directory declared. Use this file to configure the required parameters set up by means of the AsRaport system configurator and 3 different parameters:

- location of the variable definitions database (*Architect > Databases > Variable Definitions Database > Format*)
- network name of the set of alarms (*Architect > Fields and Computers > Alarms System > Alarms > Network Name*)
- name of the MS SQL server and the name of the alarm archive database (*Architect > Fields and Computers > Alarms System > Archive > Microsoft SQL Server*)

Other configuration parameters are not important. Then declare this configuration file in the *Optional configuration file* field of the AsRaport action.

It is also possible to declare report parameters.

***

The syntax of action declared manually:

ASREPORT <***report_name***>[<*optional_configuration_file*><*report_parameters*>]

Parameter ***report_name***

Meaning - name of the report created with the use of Reporting Services (the report definition stored in *.RDL file must be published on the report server of Reporting Services).

Parameter *optional_configuration_file*

Meaning - the parameter enables the declaration of XML configuration file located on other report server or in other directory than the ones declared in the configuration file of the current asix system application.

By default, information on report server and directory are retrieved from the configuration file of the current Asix system application. The configuration file enables declaration of 1 server and 1 directory on that server. Server and directory are declared using the configurator of the reporting system AsReport, available in Architect:

Architect > Application > Configure Reporting system AsReport... > Reports Browser tab

To display a report from another source (server, catalogue), you need to create a new configuration file for the Asix application with a different server or directory declared. Use this file to configure the required parameters set up by means of the AsRaport system configurator and 3 different parameters:

- location of the variable definitions database (*Architect > Databases > Variable Definitions Database > Format*)
- network name of the set of alarms (*Architect > Fields and Computers > Alarms System > Alarms > Network Name*)
- name of the MS SQL server and the name of the alarm archive database (*Architect > Fields and Computers > Alarms System > Archive > Microsoft SQL Server*)

Other configuration parameters are not important. Then declare this configuration file in the *Optional configuration file* field of the AsRaport action.

Parameter *report_parameters*

Meaning - additional report parameters.

The location of the reports displayed in the AsRapView browser is declared in the AsRapViewReportServerConfig.xml configuration file. The location refers to the place on the MS SQL Reporting Services reports server.

Asix.Evo

You can browse the reports created in the MS SQL Reporting Services directly in the Asix.Evo application by means of the report browser launched by means of the AsreportDisplay operator's action.

Intended Use

The action is used to open the window for browsing reports created in the Reporting Services technology with the use of the Asix package AsRaport module. The window allows you to generate and review selected reports. The generation parameters are pre-defined in an action or be defined interactively.

To use this action, set up the reporting services server access parameters. These parameters are provided in the Workstation settings operating panel in the Data sources of the AsixEvo.exe program.

Apart from the parameters provided in the action content, the accessibility of some of of the browser window function is also determined by the privileges (role) of the currently logged-in user.

Syntax

AsreportDisplay (*nazwa_raportu, parametry,pokazuj_parametry, współrzędna_X, współrzędna_Y, katalog_raportów*)

Parameters

nazwa_raportu

The parameter is used to enter the name of the report which is to be displayed in the browser window. The name must agree with the name of the report published on the

server and it can contain a relative path to the report directory. The parameter can be empty – the user will be able to select the report interactively.

Parameters

The parameter is used to define the value of the report generation parameters. The set of possible parameters is determined by the report definition. The parameter takes the form of a text composed of pairs *@nazwa_parametry=wartość_parametru* separated by the space character, e.g. *„@month=12 @section=A”*.

pokazuj_raporty

The *true/false* logic type parameter which determines whether the report parameter editing panel is to be displayed.

współrzędna_X

The parameter defines the X coordinate of the report browser (of the left top corner).

współrzędna_Y

The parameter defines the Y coordinate of the report browser (of the left top corner).

katalog_raportów

Name of the subdirectory (or the subdirectory relative path) on the report server, the content (names of reports and subdirectories) of which should be placed in the report selection field of the browser window. The value * means that all reports should be shown and the empty value means that the report selection field will not be shown.

8 Appendage 1: Setting up the Alarm Definition/Event Database

8.1 Alarm Definition Database - Configuration

NOTICE:

It is required to retrieve alarm definitions from Variable Definition Database in MS SQL or MDB format for the purpose of the program AsAlarm, AsTrend as well as AsRaport. In practice, the alarm definition database should be configured and alarm definitions should be joined to an existing Variable Definition Database. **It applies both to** Evo and classic **Asix application**.

The next chapters discuss the methods used to prepare the alarm definition database to work with the program AsAlarm / AsTrend / the AsRaport environment: for the classic version and for the Evo version of the Asix system.

Asix - classic version

By default, alarm definitions can be found in the text file alarm.def (or <***application_XML_file_name***> .adf). It is possible to use a different file name by defining parameter ***File Name*** (configuration of an application XML file with the use of Architect: Architect > ***Fields and Computers*** > ***Alarms System*** mode > ***Alarms*** tab / ***Alarms Base*** on the assumption that the operating mode of alarm definition database is based on text files of alarm definitions: Architect > ***Databases*** > ***Alarm Definitions Base*** > ***Type*** tab > ***Work Mode of Alarm Definitions Database Generator***: ***Alarm Definitions Base Is Generated from Text Files with Alarm Definitions***). Descriptions of alarms can also be included in several definition files. Internally alarms are identified by a number. You can define 65535 different alarms. There is no need for sequential numbering alarms. For example, numbering of one alarm group begins with the number 1000 and the second (of different origin) with the number 30000. Note that renumbering alarms in the operating system makes it necessary to delete archival files.

Definition of alarm groups can be found in **the text file group.def** (or **<*application_XML_file_name*>.gdf**). You can use the file (files) with a different name by defining the parameter ***File Name*** (configuration of an application XML file with the use of Architect: Architect > *Fields and Computers* > *Alarms System* mode > *Alarms* tab / *Alarms Base - Groups* on the assumption that the operating mode of alarm definition database is based on text files of alarm definitions: Architect > *Databases* > *Alarm Definitions Base* > *Type* tab > *Work Mode of Alarm Definitions Database Generator*: *Alarm Definitions Base Is Generated from Text Files with Alarm Definitions*).

See more:

'Asix - Functionality and Operating Rules' (Asix.pdf/chm, chapter '12. Alarm System';

'Architekt - Interactive Application Configuration Environment. Handling of VarDef - Variable Definition Database' (Architect.pdf/chm), chapter '3.9. Configuration of Alarm System'.

Asix - EVO version

See more:

Asix.Evo', ***Asix.Evo – Alarm System (PDF/CHM).***

8.1.1 Generation of Alarm Definition Database from a Spreadsheet (Classic Asix)

To add alarm definitions and alarm group definitions to variable definition database:

1. Prepare the alarm definitions in an Excel spreadsheet

> **NOTE**: Adding alarm definitions with variable definition database requires you to create an Excel spreadsheet with the definitions of alarms and the second sheet with alarm groups defined**.**

2. Set the appropriate mode of alarm definition database**:**

Architect > *Databases* > *Alarm Definitions Base* > *Type* tab > *Alarm Definitions Base Is Generated from Spreadsheets Containing Alarm Definitions* option

3. Add Excel sheets with the alarm and alarm groups definitions to sources of alarm definition database:

Architect > *Databases* > *Alarm Definitions Base* > *Data Sources* > option: *Add Data Source**:** Excel...*

4. Determine whether the alarm definitions database is to be attached to an existing variable definition database

Architect > *Databases* **>** *Alarm Definitions Base* **>** *Data Sources* **>** option: ***Add Alarm Definitions Base to Variable Definition Base***

5. Generate alarm definition database (this operation adds alarm and alarm group definitions to an existing variable definition database)**:**

Architect > *Databases* > *Alarm Definitions Base* > *Data Sources* > *option:* ***Operations***: Generate.

NOTE: Each update **of** variable definition database (the re-generation) requires re-generation of alarm and alarm group definitions to the variable definition database using the following command:

Architect > *Databases* > *Alarm Definitions Base* > *Data Sources* > *option:* ***Operations***: Generate.

8.1.2 Collective Alarm and Alarm Group Definition Database Generated from Many Applications (Classic Asix)

Generator of collective alarm and alarm group definitions will create a collective alarm definition database based on sources defined in the specified applications. Created database is added to the variable definition database of currently parameterized application.

To generate a collective alarm definition database from many applications:

1. Set the appropriate mode of alarm definition database operation**:**

Architect > *Databases* > *Alarm Definitions Base* > *Type* tab > *Collective Alarm Definitions Base is Generated from an Alarm Definitions Database of Other Applications* option

2. Add data sources:

Architekt > *Databases* > *Alarm Definitions Base* > *Data Sources* tab > option: ***Add Data Source:*** INI files or XML files

3. Generate alarm definition database (this operation adds alarm and alarm group definitions from many applications to an existing variable definition database of parameterized application)**:**

Architect > Databases > Alarm Definitions Base > Data Sources > *option:* ***Operations****:* ***Generate****.*

8.1.3 Alarm Definitions Base Generated from Text Files

For classic applications, alarm definitions are stored in .adf (or alarm.def) files usually, and alarm group definitions in .gdf (or group.def) text files.

To add alarm definitions and alarm group definitions from text files to variable definition database:

1. Prepare alarm definitions in text files.

   Information on how to create such files can be found in the manual: *Asix.PDF/CHM, 12. Alarm System*.

2. Set the appropriate mode of alarm definition database:

Architect **>** *Databases* **>** *Alarm Definitions Base* **>** *Type* tab **>** *Alarm Definitions Base Is Generated from Text Files with Alarm /Definitions*

3. Declare text file name:

Architect **>** *Fields and Computers* **>** *Alarms System* **>** *Alarms* **>** *Alarms Base* / *Alarms Base - Groups* tab

4. Choose *Field/Computer* for which text files were declared in 3 step. It will automatically load files of alarm and alarm group definitions to the generator window:

Architect > *Databases* **>** *Alarm Definitions Base* **>** *Data Sources* tab > option: *Field/Computer*

5. Generate alarm definition database (this operation adds alarm and alarm group definitions to an existing variable definition database of parameterized application):

Architect > Databases > Alarm Definitions Base > Data Sources > *option:* ***Operations: Generate****.*

8.1.4 Alarm Definition Database Generated based on SQL Alarms (Classic and Evo Asix)

Generating the alarm definition database based on SQL alarms of EVO applications refers to the case when information on alarm events are stored not only in archival alarm logs but also additionally in alarm archive of SQL type. In the same way is in the case of classic applications - where archiving in MS SQL database may be carried out in parallel with archives stored in disc files. Generator (in Architect) of SQL alarm definition database generates alarm definition database added to variable definition database of Asix application - all this on the basis of event archive in SQL database.

To add SQL alarm definition to variable definition database:

1. Determine the source of alarm definitions: (it could be the Asix application directory in EVO version, XML files in classic Asix, INI files of older versions of classic Asix applications as well as a current application):

Architect > *Databases* > SQL *Alarm Definitions Base* **>** *Data Sources* **>** option: ***Add Data Source***

2. Generate alarm definition database (this operation adds alarm and alarm group definitions to an existing variable definition database of parameterized application)**:**

Architect > Databases > SQL Alarm Definitions Base > Data Sources > option: ***Operations: Generate****.*

8.2 Alarm Event Database - Configuration

NOTICE:

Archiving alarm events in SQL type database is required for the AsAlarm program and AsReport reporting environment.

Classic Asix

The current status and alarm history of Asix system are stored in disk files. Alarm files are stored in subdirectory 'alarms' of the startup directory (if it is not declared a different location). All active alarms are stored in a file named 'alarms.act' . Alarm history is stored in a set of files with names derived from the pattern al??????.log. In each such *a* file the alarms of one day are stored. Date of that day *is* encoded in the place of ?????? characters (day, month and year). Alarm history files can be stored with no fixed time. You can also limit the amount of stored days.

There is the possibilty to convert binary files with alarm event logs of Asix system *(*al??????.log files) into the format of SQL database.

It is also possible the current recording of alarm event in SQL database. This requires adequate configuration of the XML file of an Asix system application while unlocking writing the data on alarm events generated on-line into an SQL database.

(see: - 8.2.1 Starting Recording Alarm Events together with Conversion of Binary Files to SQL Database (Classic Asix) - when there is the need for continuity in the registration of alarm events between the archives and the current registration

- 8.2.2 Starting On-line Registration of Alarm Events (Classic Asix) - when there is no alarm archive and registration starts from the surrent moment)

EVO Asix

How to set up recording of alarm events in an SQL database for EVO application describes the chapter *8.2.3 Archiwum alarmów SQL.*

8.2.1 Starting Recording Alarm Events together with Conversion of Binary Files to SQL Database (Classic Asix)

Conversion of binary files of Asix alarm archive into an SQL database is implemented via AlarmLogConverter.exe which is normally attached to the Asix package.

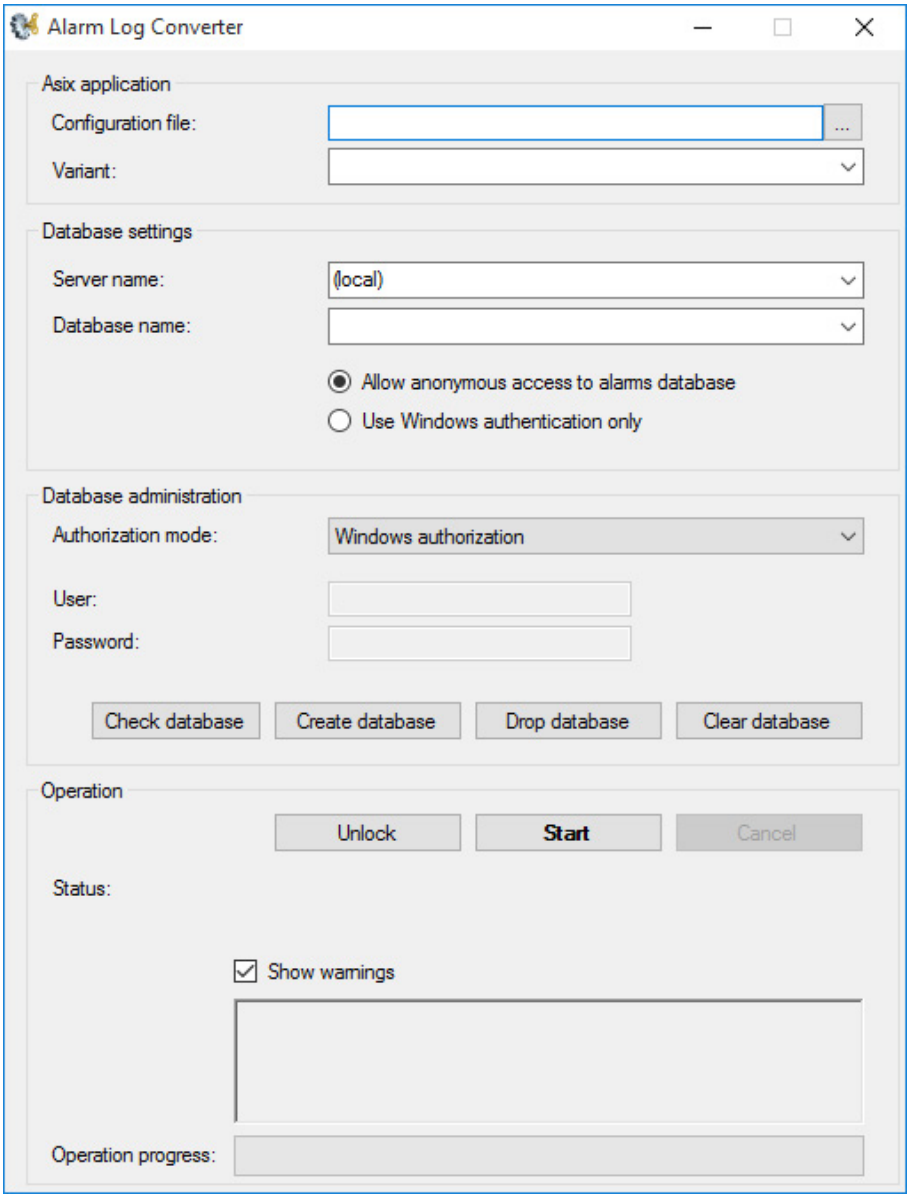

Fig. Main Window of AlarmLogConverter.

Options required to start recording alarm events and perform conversion:

Asix Application – it refers to the source of the alarm

Configuration File – it requires Asix application configuration file: XML for Asix 5-6 and classic 7 or INI file for older wersions

Variant – application variant dedicated for a special field / computer (a few computers operating under a given application at the factory)

Database Settings – parameters concerning locations and target database of alarms

Server Name – MS SQL Server name

Database Name - name of MS SQL database of event alarms

Allow Anonymous Access to Alarms Database / Only Windows Authentication Only – determine how to access alarm database

Database Administration – includes options and commands related to administration of databases

Authorization Mode - specify the way of authorization of SQL server database administrator

Check Database – used to check whether the database exists and verify its correctness

Create Database - used to create a new alarm database

Drop Database – used to remove a database

Clear Database – used to clear database in order to re-convert

Operations – commands directly related to conversion procedure

Unlock – unlock recording data on alarm events generated on-line into SQL database from the moment of switching on the option EXPORT_CHANNEL which activates creation of buffer files, with data about these events, by Asix. Unlock saves data stored in buffer files into SQL database.

Start – start conversion of al??????.log alarm files to the MS SQL Server database format

Cancel – interrupt the conversion (conversion is not completed at all)

 Procedure of start-up of recording alarm events with conversion of binary files to SQL database format:

To convert al??????.log files to SQL database format and to start up the current registration of alarm events without loss of data on instances of alarms during conversion - proceed strictly according to the following steps:

1. Create an empty SQL database by using the converter AlarmLogConverter:

    a. open AlarmLogConverter

    b. declare an XML file of an application indicating the computer station from which alarm archival files will be retrieved

    c. enter SQL server name and alarm target database name

    d. specify the authentication mode for utility and administration functions

    e. click ***Create Database***

2. Declare the current registration of alarm events (until you unlock recording into MS SQL database, these events will be recorded in buffer files):

    a. open the current Asix application configuration file using Architect

    b. in Architect > *Fields and Computers* **>** *Alarms System* **>** *Archive/Microsoft SQL Server* tab declare options:

    Name of Microsoft SQL Server - name of MS SQL Server server database of alarm events,

    Database Name - name of database of alarm events.

    c. Save all changes of the application configuration file

3. Restart the application. Since that moment events are saved in buffer files.

4. Conversion of al??????.log files to SQL database format:

    a. go to AlarmLogConverter and click ***Start***.

5. Unlock recording alarm events generated on-line to SQL database:

    a. after the completion of the convertion click ***Unlock*** (in AlarmLogConverter); the operation rewrite alarm events from buffer files to SQL database - included only the events not covered by the conversion of al??????.log event files

8.2.2 Starting On-line Registration of Alarm Events (Classic Asix)

Starting alarm registration of Asix system to an SQL database is realised via AlarmLogConverter.exe which is included in the Asix package by default.

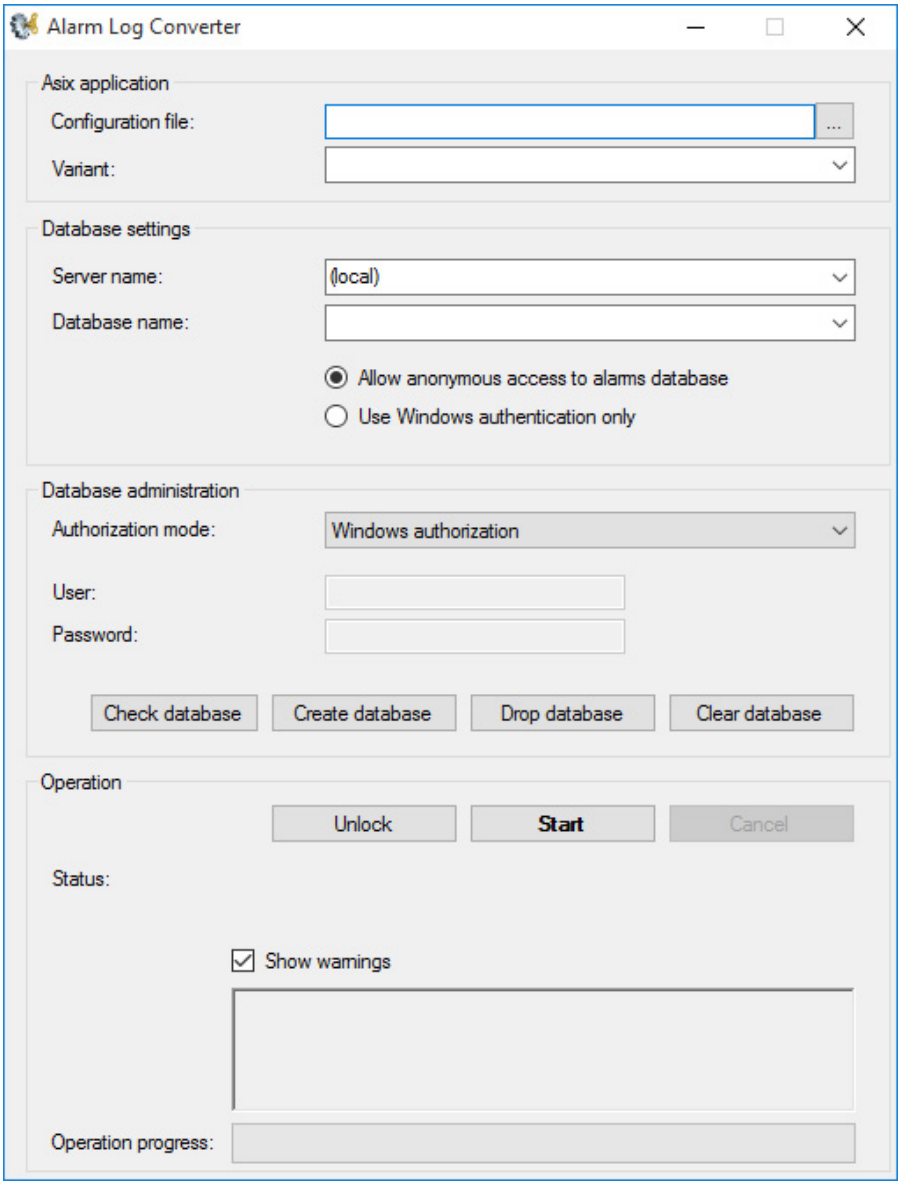

Fig. Main Window of AlarmLogConverter.

Options required to start recording alarm events:

Asix Application – it refers to the source of the alarm

> ***Configuration File*** – it requires Asix application configuration file: XML for Asix 5-6 and classic 7 or INI file for older wersions

Variant – application variant dedicated for a special field / computer (a few computers operating under a given application at the factory)

Database Settings – parameters concerning locations and target database of alarms

Server Name – MS SQL Server name

Database Name - name of MS SQL database of event alarms

Allow Anonymous Access to Alarms Database / Only Windows Authentication Only – determine how to access alarm database

Database Administration – includes options and commands related to administration of databases

Authorization Mode - specify the way of authorization of SQL server database administrator

Check Database – used to check whether the database exists and verify its correctness

Create Database - used to create a new alarm database

Drop Database – used to remove a database

Clear Database – used to clear database in order to re-convert

Operations – commands directly related to conversion procedure

Unlock – unlock recording data on alarm events generated on-line into SQL database from the moment of switching on the option EXPORT_CHANNEL which activates creation of buffer files, with data about these events, by Asix. Unlock saves data stored in buffer files into SQL database.

Start – start conversion of al??????.log alarm files to the MS SQL Server database format

Cancel – interrupt the conversion (conversion is not completed at all)

 Procedure of start-up of recording alarm events to SQL database format (without conversion of binary files):

To start up the current registration of alarm events - proceed strictly according to the following steps:

1. Create an empty SQL database by using the converter AlarmLogConverter:

    a. open AlarmLogConverter

    b. declare an XML file of an application indicating the computer station from which alarm archival files will be retrieved

    c. enter SQL server name and alarm target database name

    d. specify the authentication mode for utility and administration functions

    e. click ***Create Database***

2. Declare the current registration of alarm events (until you unlock recording into MS SQL database, these events will be recorded in buffer files):

    a. open the current Asix application configuration file using Architect

    b. in Architect > *Fields and Computers > Alarms System > Archive/Microsoft SQL Server* tab declare options:

    Name of Microsoft SQL Server - name of MS SQL Server server database of alarm events,

    Database Name - name of database of alarm events.

    c. Save all changes of the application configuration file

3. Restart the application. Since that moment events are saved in buffer files.

4. Unlock recording alarm events generated on-line to SQL database:

a. after the completion of the convertion click *Unlock* (in AlarmLogConverter); the operation rewrite alarm events from buffer files to SQL database.

8.2.3 Configuration of Recording Alarm Events in SQL Database (EVO Asix)

The SQL type alarm archive for Asix.Evo is an additional solution for the standard historical alarm log. This archive is designed to analyze alarms with the use of AsAlarm as well as in AsReport reporting system. In order to improve reliability, you can create archives on multiple servers simultaneously. However, even when using a single server, Asix.Evo has mechanisms of data buffering, that allows for operation with temporary loss of connection with SQL server. The SQL archive should be parameterized in the same way on all stations of a controller type. It is because writing to the SQL database is performed at a given moment only by the station of an active controller.

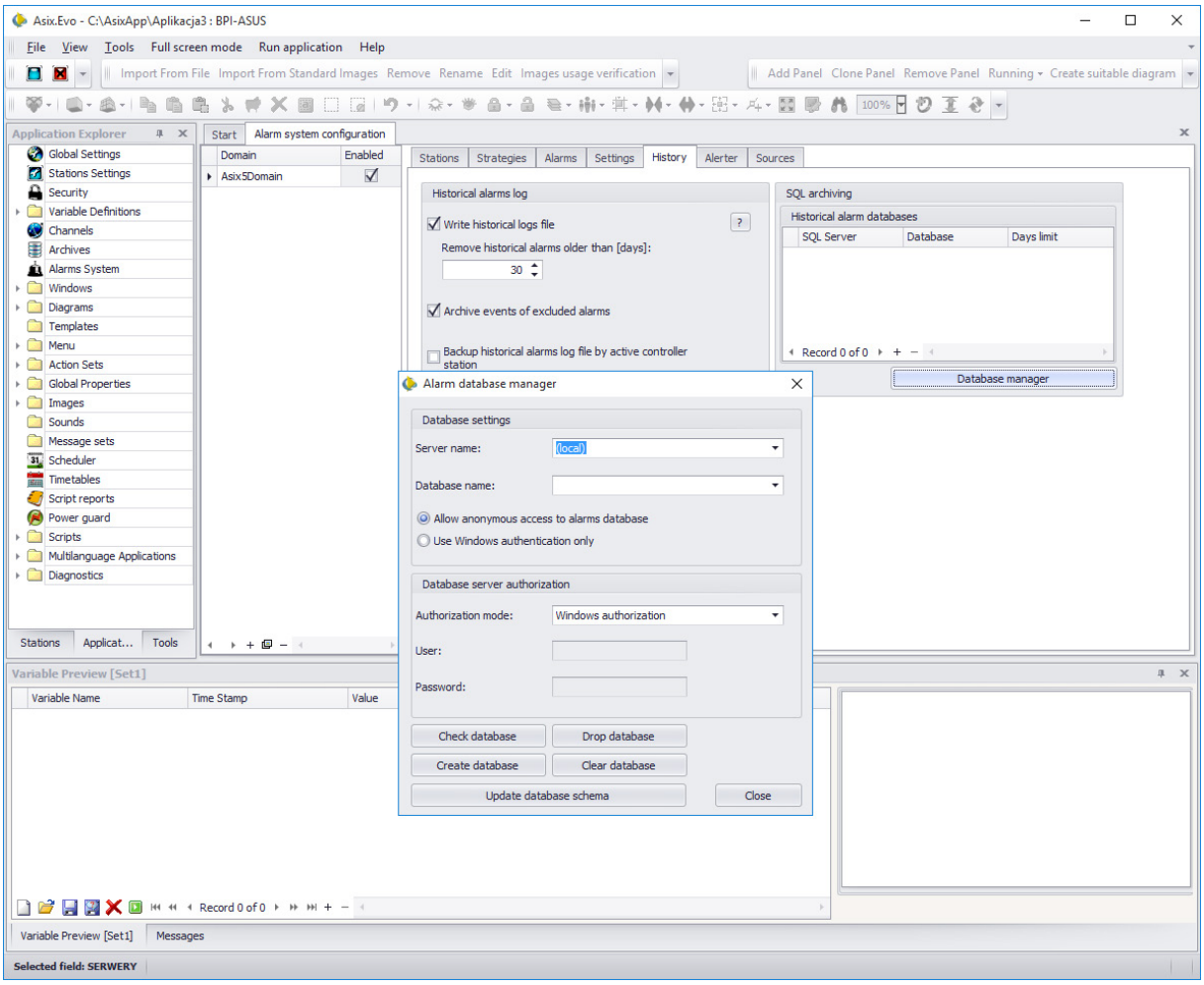

Fig. AsixEvo.exe > 'Alarms System' > 'Alarm System Configuration' Tab > 'History' Tab > SQL Archiving > Database Manager.

Before starting archiving SQL Alarm Database must be created. It is done by the window above opening by the button *Database Manager*. The window allows also for performing other administrative tasks. The credentials from the group Database Server Authorization must ensure authorizations in the SQL server sufficient to perform operations of database creation and dropping.