



BridgeWorX – Using the Text File Parser Block



APPLICATIONS NOTE

October 2013

Description: Guide to importing data from a text file into an OPC tag or database using BridgeWorX.

OS Requirement: Win XP/Server 2003/ Server 2008/Vista/Windows 7

General Requirement: BridgeWorX version 9.x must be installed.

Introduction

BridgeWorX offers a large set of data integration blocks. The Text File Parser block discussed in this article can be used to take data from a text file and transform it ready to be used in OPC tags or database tables.

The Text File Parser block will read data from a text file and uses various user configurable properties to determine how the data will be transformed as part of this process. This configuration example shows how some of these properties can be used to interpret a text file and be transformed, after which the resultant data will be inserted into a database table.

Example Text Data File

The following section defines the file that will be read by the Text File Parser block. This file is built up out of two sections: a header section and data section. It is not necessary to have a header section, but to show the abilities of the Text File Parser block it is included in this example text file. The text file also includes a row with column information for the data. Again this is not necessary for the Text File Parser block to work; however, it will show the range of capabilities of the block.

[Header Info]

This file has some additional information in a header, before there is a data section containing the actual values that need to be read. The values are preceded by a row with header information

[Data Values]

Column1,Column2,Column3,Column4

1.23,4.56,7.89,9.102

-12.23,1e10,-12e-2,109.2

289.273,200,200,2000000000

12.12,13.13,14.14,0

1,0,,19

Example Database Table

This example exports data to a database table in SQL Server. In this case the table has four columns defined, each of data type float.

VTC-WS2-910\...svDestination			
Summary			
Column Name	Data Type	Allow Nulls	
Column1	float	☒	
Column2	float	☒	
Column3	float	☒	
Column4	float	☒	
		☐	

Figure 1 – The database table used for the example

BridgeWorX Database Connector

Before the transaction can be created a BridgeWorX database connector has to be created with a “Writing to Database” action defined. The action will have an operation type of **Inserting record(s) into a table** and point to the table defined in your database server. Name this connector **CsvDestination**.

BridgeWorX Transaction Diagram

With the text file available and the database connector now created the transaction diagram can be assembled. Create a new transaction diagram and insert the Text File Parser block, which can be found under the Integration Tasks group in the Transaction Designer. Double-click on the just inserted block to open the property pages for the block. Select the Text File Parser Properties tab and set the “Operation to execute” to **ParseTextFile** and set the Destination Type to **Database**.

With the basic properties defined click on the More Properties button to define the file to be parsed and how it should be parsed. Use the Source Directory and Source File(s) fields to specify which file(s) need to be parsed by this block. After the files have been defined the properties for processing the files have to be defined.

1. **Data Column Separator:** specifies the character that will be used to identify the columns of data. For the test file defined earlier this will be a comma.
2. **Data Encapsulation Character:** specifies the character that will be used to encapsulate data column values if the value may contain the character used for separation. This is commonly used if string data has to be imported and the string data may contain the separation character, e.g. commas. For the test case this will be set to none as no string text will be imported.
3. **Column Header Options:** If data column headers are defined inside the source file they can be specified here. You can define both where in the file the header line is located, and how many lines it occupies. For the test case the Header Line Offset will be set to 7 as column headers (Column1 – Column4) are located at this line. And the

Number of Lines will be set to 1 as all headers are on a single line.

4. **Data Processing Options:** If data does not start directly at the top or if each data line is spread across multiple lines this can be configured in this section. This section will also give you the option to extract a very specific set of columns rather than all data. For the test case the Data Line Offset will be set to 8 as the data starts at this line. The Number of Lines will be set to 1 as all data for a row is contained on a single line. The Maximum number of columns to retrieve will be set to 4 as this is number of data columns in the test file. The First column offset will be set to 0 as all data columns will be used.

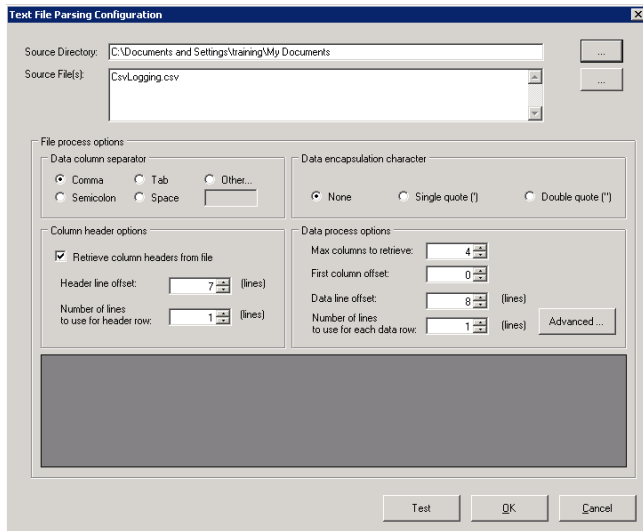


Figure 2 – Text File Parser block configuration

The Test button can be used to visualize the results of the configured parameters on the specified file. Click on OK to complete the configuration of the parsing options.

Once back at the properties page select the Return Values Mapping tab and select the previously created CsvDestination database connector and map all text file input columns (Column1 – Column4) to the database columns (Column1 – Column4). The Text File Parser block is now fully configured.