The CSVRead Driver

The CSVRead driver allows North to read delimiter-separated text files, and provide access to their values using objects. Any delimiter character may be used to separate the values in a file, providing support to read comma-separated value files (CSV), tab-separated value files (TSV), and initialisation files (INI). Available for Commander and ObSys.

This document relates to CSVRead driver version 1.0

Please read the *Commander Manual* or *ObSys Manual* alongside this document, available from *www.northbt.com*

Contents

Purpose of CSVRead Driver

The CSVRead driver allows North to read delimiter-separated text files, and provide access to their values using objects. Any delimiter character may be used to separate the values in a file, providing support to read comma-separated value files (CSV), tab-separated value files (TSV), and initialisation files (INI).

The driver can read values from up to 32 different files.

CSVRead can access files stored remotely on a web server using HTTP/WebDAV. Authentication is not supported.

On ObSys, CSVRead can also access files stored on the local PC or network path.

Two related drivers are also available. The XmlRead driver can read data from an xml file, and the CSVWrite driver can write values to a delimiter-separated text file.

Values

From a file, the driver can access up to 32 columns of data. Each column can contain one of the following value types:

- Short Text – string with maximum length of 31 characters
- Long Text – string with maximum length of 127 characters
- Number – positive integer
- Float – signed floating-point number with up to 4 decimal places
- Ignore – discard the data in the column

Prerequisites

Locate the files that the driver should read. Determine the delimiter character – typically a comma for CSV files. Find the columns within the files that are of interest. Be aware that when editing or viewing a file, some software locks the file so that other applications cannot access it.

Operation

Cache

The driver can access data from up to 32 files, but can only hold in memory the data from four files at any one time. Therefore, the driver uses a caching mechanism.

When a task requests an object from the driver for a value within a particular file, the driver loads the file into memory. If necessary, the driver will overwrite a file already in memory. Once loaded, the driver holds the file in memory until the file's data is deemed out-of-date (i.e. its cache life has expired), or until the memory is needed for another file.

If the driver needs to overwrite a file already in memory, it keeps the most recently accessed files and overwrites the file that has not been accessed for the longest period.

Caution

If data from more than four files is required, care should be taken to ensure tasks requesting values from the files are configured to allow the caching system to work efficiently.

- Consider the cache life for each file carefully
- Tasks should request all the data required from one file before requesting from another
- If several tasks need data from files, consider loading the data into Essential Data, and have the other tasks access the data from there.

File Size

The driver allocates up to 24kB (24,576 bytes) of memory for each cached file. This memory can hold up to 1000 rows of up to 32 columns of data. The actual maximum number of rows that can be cached will depend on how each column type is configured.

To maximise the number of rows available, and access larger files:

- Configure each column type with the best matching format

Column Type	Usage	Memory Used (bytes)
Short Text	Text field, up to 31 characters	32
Long Text	Text field, up to 127 characters	128
Number	Unsigned integer number	4
Float	Floating-point number with a decimal point	4
Ignore	Data in the column is not stored	0

- Set the column type to 'Ignore' when its data will not be used
- If required, collect the same file twice with different columns for each file setup.

If necessary, it is possible to calculate whether a file can be cached in its entirety: combine the memory needed for each row of the file, and multiply this by the row count. Alternatively, Maximum rows in cache (MR) and Rows in file (RC) objects are available.

Example CSV File

The example below shows a comma-separated variable file.

Example.csv

```
Name,Code,Level
Floor 1,D5476,21.5
Floor 2,G5164,22.2
Floor 3,W4652,25.2
Floor 4,F4537,21.6
Floor 5,B3518,20.7
```

The file has a header row, which contains labels for the subsequent rows of data. Each data row contains three items of data: a short text field called 'Name', a short text field called 'Code', and a floating-point value called 'Level'.

Using the Driver

On ObSys and Commander, the CSVRead driver is pre-installed. Once started, you will need to set up the driver before it can read data from delimiter-separated files.

Starting the Interface

⌨ To start an interface using the CSVRead driver, follow these steps:

→ **Start Engineering** your North device using ObSys

→ Navigate to **Configuration, Interfaces,** and set an unused **Interface** to 'CSVRead' to start the particular interface

→ Navigate to the top-level of your North device and re-scan it

The driver setup object (M*c*), labelled **CSVRead Setup,** should now be available.

Setting up the Driver

⌨ To set up the driver, follow these steps:

→ Navigate to the **CSVRead Setup** object (M*c*). For example, if you started interface 1 with the driver earlier, then the object reference will be 'M1'

→ Navigate to **File 1** (F1) and set the **Server IP/DNS** (SN), **Filename** (FN), **Cache life (mins)** (CL), **Header rows** (HH) and **Delimiter (ASCII Code)** (SC)

→ Set the **Column type** for each column of data required

→ Repeat for each file.

Object Specifications

Once an interface is started, one or more extra objects become available within the top-level object of the device. As with all North objects, each of these extra objects may contain sub-objects, (and each of these may contain sub-objects, and so on) - the whole object structure being a multi-layer hierarchy. It is possible to navigate around the objects using the ObSys Engineering Software.

Each object is specified below, along with its sub-objects.

Device Top-Level Objects

When an interface is started using the CSVRead driver, the objects below become available within the top-level object of the device. For example, if interface 1 is started, then the object reference 'M1' becomes available.

Description	Reference	Type
CSVRead Setup Set up the CSVRead driver, started on interface c (c is the interface number)	Mc	Fixed Container: On the Commander platform this will be *[CDM v20\CSVRead v10]* On the ObSys platform this will be *[OSM v20\CSVRead v10]*
File Values	Sc	Variable Container: *[CSVRead v10]*

CSVRead Setup

The CSVRead driver contains the following objects.

Description	Reference	Type
System Label Label displayed when scanning the system	DL	Obj\Text; Max 20chars; Adjustable
File *x* Configuration details for reading File *x*, where *x* is in the range 1..32	F*x*	Fixed Container: On the Commander platform this will be *[CDM v20\CSVRead v10\File]* On the ObSys platform this will be *[OSM v20\CSVRead v10\File]*
Debug Enable This will store additional debug information in the record file. Use this option only when instructed by North Support	DE	Obj\Num; Adjustable

CSVRead File

Object Type: [OSM v20\CSVRead v10\File]
Object Type: [CDM v20\CSVRead v10\File]

A CSVRead File contains the following objects.

Description	Reference	Type
Server IP/DNS DNS or IP address of remote web or WebDAV server, with TCP port number (default 80). On ObSys platform, leave this field blank to accessing a local PC or network path	SN	Obj\Text; Max 120 chars; Adjustable Format: <DNS name or IP address>:<port> Example: data.northbt.com:8080
Filename Resource name of file to read. On ObSys, for a local PC or network file this should include the full path. To assist in reading a time-based filename, the following variables can be used: $(th) – current time: hours (2-digit) $(tm) – current time: minutes (2-digit) $(ts) – current time: seconds (2-digit) $(dd) – today's date (2-digit) $(dm) – today's month (2-digit) $(dy2) – today's year (2-digit) $(dy4) – today's year (4-digit) $(yd) – yesterday's date (2-digit) $(ym) – yesterday's month (2-digit) $(yy2) – yesterday's year (2-digit) $(yy4) – yesterday's year (4-digit) $(obs) – ObSys data folder $(doc) – User's Documents folder	FN	Obj\Text; Max 127 chars; Adjustable Examples: assets\data.csv C:\Users\ABC\Downloads\Data.csv $(obs)\CSV\Meter5_$(dy4)(dm)(dd).csv
Cache life (mins) Once a file is loaded, it will be held in memory for the cache life duration before requiring a re-read. Refer to *Operation*.	CL	Obj\Num: 0…60; Adjustable
Header rows The number of rows at the start of the file that should be ignored	HH	Obj\Num; Adjustable
Delimiter (ASCII Code) ASCII code of the character used to separating information in columns of the file	SC	Obj\Num: 32…126; Adjustable The ASCII code for a comma is 44, tab is 9, equal is 61
Column *x* type Describes the type of information in column *x*. If set to Ignore, then column data will not be stored. Column, *x*, is in the range 1…32	C*x*	Obj\ENum: 0…4; Adjustable Values: 0=Ignore, 1=Short Text (up to 31 chars), 2=Num (unsigned), 3=Float, 4=Long text (up to 127 chars)
Maximum rows in cache Shows the number of rows in the file that the driver can cache, based on the configured column types. Refer to *Operation*.	MR	Obj\Num: 0…1000
Cache status Indicates if the driver has cached the file. 'Not read' indicates that an initial request for a value has not yet been made, or the file has failed to load	S	Obj\Enum: 0…2 Values: 0=Not read, 1=Reading, 2=Cached

Description	Reference	Type
Rows in file Total number of rows contained within the file. Refer to object MR for number of rows that will be cached	RC	Obj\Num

File Values

Object Type: [CSVRead v10]

The File Values object contains data from up to 32 files, configured in the driver.

Description	Reference	Type
File *x* The file number, *x*, can be in the range 1…32	F*x*	Variable Container: *[CSVRead v10\File]*

File

Object Type: [CSVRead v10\File]

A File contains the column data from within the file.

Description	Reference	Type
Column *x* The column number, *x*, can be in the range 1…32	C*x*	Fixed Container: Type depends on how column type is defined in driver setup. *[CSVRead v10\ShortTextColumn]* *[CSVRead v10\LongTextColumn]* *[CSVRead v10\NumColumn]* *[CSVRead v10\FloatColumn]*
Label Filename configured in driver setup	L	Obj\Text

Short Text Column

Object Type: [CSVRead v10\ShortTextColumn]

The Column object contains the row values from a particular column in the file.

Description	Reference	Type
Row _x_ Row, _x_, can be in the range 1…1000	R_x_	Obj\Text; Maximum 31 chars

Long Text Column

Object Type: [CSVRead v10\LongTextColumn]

The Column object contains the row values from a particular column in the file.

Description	Reference	Type
Row _x_ Row, _x_, can be in the range 1…1000	R_x_	Obj\Text; Maximum 127 chars

Number Column

Object Type: [CSVRead v10\NumColumn]

The Column object contains the row values from a particular column in the file.

Description	Reference	Type
Row _x_ Row, _x_, can be in the range 1…1000	R_x_	Obj\Num: 0…4,294,967,295

Float Column

Object Type: [CSVRead v10\FloatColumn]

The Column object contains the row values from a particular column in the file.

Description	Reference	Type
Row _x_ Row, _x_, can be in the range 1…1000	R_x_	Obj\Float

Driver Versions

Version	Build Date	Details
1.0	23/03/2015	Driver released
1.0	06/04/2016	Added objects MR, S, and RC to CSV Read object

Next Steps…

If you require help, contact support on 01273 694422 or visit *www.northbt.com/support*

North Building Technologies Ltd
+44 (0) 1273 694422
support@northbt.com
www.northbt.com

Author: GS
Checked by: JF

Document issued 08/09/2022.