Logo

Description automatically generated

**GAP Plug-in Configuration Guide**

Contents

[Introduction 2](#_Toc119310985)

[Supported Collection Types 2](#_Toc119310986)

[Data Sources 2](#_Toc119310987)

[CSV via File System 3](#_Toc119310988)

[Server Properties 3](#_Toc119310989)

[Field Definitions 3](#_Toc119310990)

[CSV via Http 5](#_Toc119310991)

[Server Properties 5](#_Toc119310992)

[Field Definitions 5](#_Toc119310993)

[Microsoft SQL 7](#_Toc119310994)

[Server Properties 7](#_Toc119310995)

[Field definitions 8](#_Toc119310996)

[ODBC 9](#_Toc119310997)

[Server Properties 9](#_Toc119310998)

[Field Definitions 10](#_Toc119310999)

[Available GAP values 11](#_Toc119311000)

[Current Plugin 11](#_Toc119311001)

[Job Messages 13](#_Toc119311002)

[Legacy GAP Plugin 14](#_Toc119311003)

[Examples 15](#_Toc119311004)

[CSV File 15](#_Toc119311005)

[SQL Query 16](#_Toc119311006)

[Troubleshooting 18](#_Toc119311007)

[Reporting Notes 18](#_Toc119311008)

[Appendix A - Upload legacy plugin historical data to a new backup server 18](#_Toc119311009)

[Appendix B - Example to HTTPS Configuration for Testing 19](#_Toc119311010)

[Technical Support 27](#_Toc119311011)

# Introduction

This document describes the Bocada GAP Plug-in. The GAP Plugin is primarily used to bring backup job data into the Bocada database for reporting when a backup product-specific plugin is not available. Backup data can be collected via a CSV file, through an ODBC connection, or Microsoft SQL Server.

In some production applications, the GAP plugin is used for situations in which the backup server cannot be reached by the normal Bocada Data Collection Service, but when the data can be shipped out of the backup server in a controlled CSV format.

# Supported Collection Types

The plug-in currently supports the following collection types:

|  |  |  |
| --- | --- | --- |
| **Collection Type** | **Supported** | **Description** |
| Backup | ✓ | Collects transactional details about backup, duplication, and restore jobs. Example metrics include start times, durations, bytes, files, errors, etc. |
| Storage |  | Collects point-in-time inventory information. Example metrics include total recoverable gigabytes (storage), media volume count, media volume status, etc. |
| Policy |  | Collects and stores information on policy attributes, schedules, storage units, storage groups, storage lifecycle policies, and clients. |

# Data Sources

The GAP Plug-in works similarly to other Bocada Plug-ins, data is collected by a Bocada Data Collection Server.

The plugin relies on any of the following data sources:

* **CSV via File System**: pull data from a CSV text file. The data is mined from a comma-separated-value CSV data file.
* **CSV via HTTP**: pull data from a CSV file accessed via an HTTP or HTTPS path.
* **Microsoft SQL**: integrated support with Windows or SQL authentication.
* **ODBC driver**: with customizable configuration strings and templates.

##### Legacy Plugin

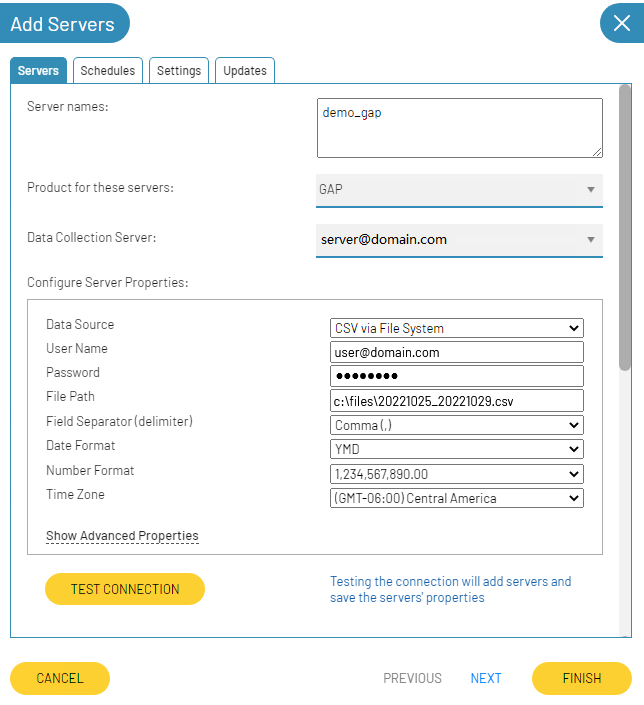
The GAP plugin supports the legacy "Manual" plugin, so it allows the upload of CSV Files in the Manual plugin format or with the extended new format.

# CSV via File System

The GAP plug-in mines data from a comma-separated value (CSV) format.

The list of available data fields that can be imported is in the section below named Available GAP values.

### Server Properties



### Field Definitions

##### Server name

This name can be anything but name the server so that is clear to which instance the data belongs.

##### Data Source

Specify the kind of data source you are using to extract the data. Select “CSV via File System” to load the data from a ***csv*** file, either located on the machine or on a network drive.

##### User Name / Password

For windows authentication, in case the credentials are required, use the format of username@domain.com or Domain\Username.

##### File Path

Specify the path location of the CSV file. You can use absolute paths or UNC, e.g., “C:\GapFiles\file.csv” or “\\machine\shared\file.csv”. The file must have the extension ***.csv***.

##### File Separator (Delimiter)

Set the delimiter of the csv file. The allowed options are:

* Comma (,)
* Semicolon (;)
* Pipe (|)

##### Date Format

Select the date format. Either “/” or “-“can be used for the date. So, you can use DD-MM-YYYY and DD/MM/YYYY, and both formats are taken as correct if you select the option “DMY”. Likewise, hours are allowed both on a 24-hour basis and a 12-hour basis.

There are three options to indicate the date format:

* YMD: indicates that the date format is YYYY-MM-DD or YYYY/MM/DD, e.g., 2022-10-18 12:07:28.
* MDY: indicates that the date format is MM-DD-YYYY or MM/DD/YYYY, e.g., 10-18-2022 16:45:26.
* DMY: indicates that the date format is DD-MM-YYYY or DD/MM/YYYY, e.g., 2022-10-18 04:45:26 PM.

To support the legacy plugin format, date without time is allowed, for example either of the following options is allowed:

* 10/31/2022 19:30:00
* 10/31/2022 7:30:00 PM
* 10/31/2022

##### Number Format

Set the number format according to the format of the numbers that the data to load has.

##### Time Zone

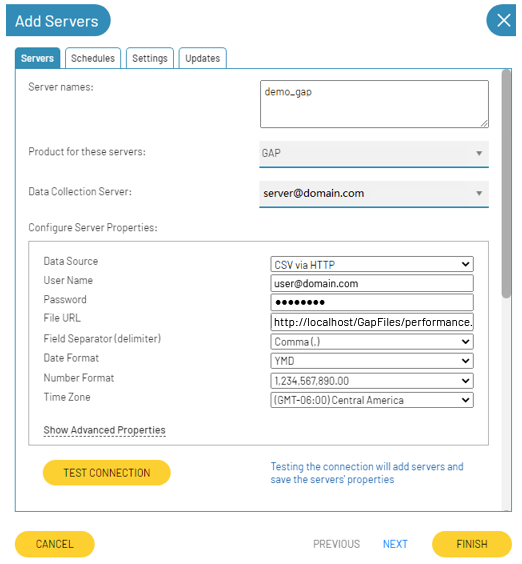
Select the time zone where the server resides. This setting ensures times are displayed consistently in environments that span multiple time zones.

# CSV via Http

The Gap plugin allows you to read CSV files that are shared via an HTTP or HTTPS URL. To configure the backup server, you must indicate “CSV via HTTP” in the Product option.

The list of available data fields that can be imported is in the section below named Available GAP values.

### Server Properties



### Field Definitions

##### Server name

This name can be anything but name the server so that is clear to which instance the data belongs.

##### Data Source

Specify the kind of data source you are using to extract the data. Select “CSV via HTTP” to load the data from a ***csv*** file located in an http URL.

##### User Name / Password

If credentials are required, you can indicate the username and password that that will be used to make the request and pull the file. When the credentials are indicated, the process will use Basic Authentication in the Http request, otherwise, basic authentication will not be used in the request.

##### File URL

Specify the URL location of the CSV file, e.g., *https://fileserver/directory/file.csv*. The file must have the extension ***.csv***.

##### File Separator (Delimiter)

Set the delimiter of the csv file. The allowed options are:

* Comma (,)
* Semicolon (;)
* Pipe (|)

##### Date Format

Select the date format. Either “/” or “-“ can be used for the date. So, you can use DD-MM-YYYY and DD/MM/YYYY, and both formats are taken as correct if you select the option “DMY”. Likewise, hours are allowed both on a 24-hour basis and a 12-hour basis.

There are three options to indicate the date format:

* YMD: indicates that the date format is YYYY-MM-DD or YYYY/MM/DD, e.g., 2022-10-18 12:07:28.
* MDY: indicates that the date format is MM-DD-YYYY or MM/DD/YYYY, e.g., 10-18-2022 16:45:26.
* DMY: indicates that the date format is DD-MM-YYYY or DD/MM/YYYY, e.g., 2022-10-18 04:45:26 PM.

To support the legacy plugin format, date without time is allowed, for example either of the following options is allowed:

* 10/31/2022 19:30:00
* 10/31/2022 7:30:00 PM
* 10/31/2022

##### Number Format

Set the number format according to the format of the numbers that the data to load has.

##### Time Zone

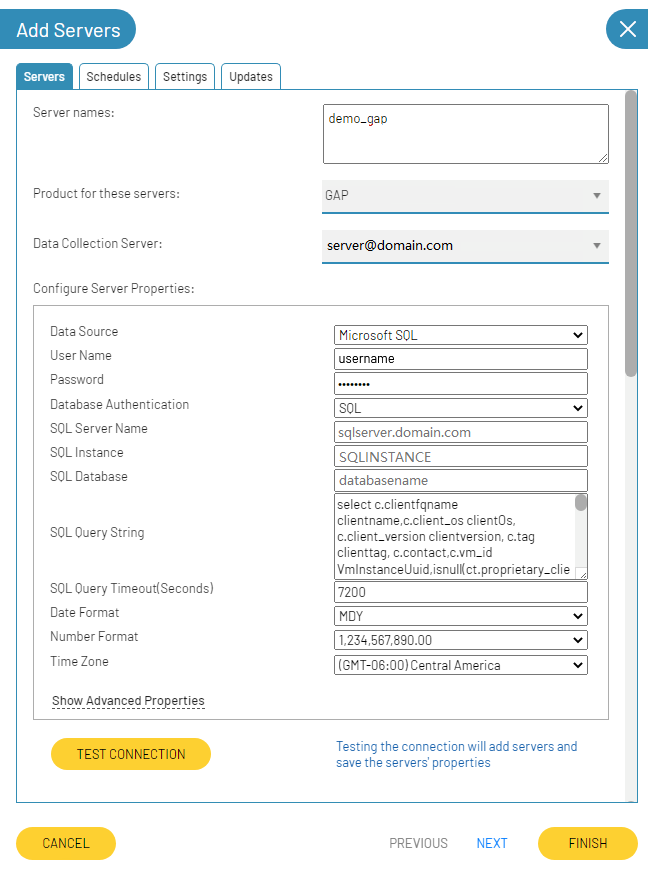
Select the time zone where the server resides. This setting ensures times are displayed consistently in environments that span multiple time zones.

# Microsoft SQL

When the backup data is stored in a Microsoft SQL database, configure your GAP plugin to extract this data from it.

The list of available data fields that can be imported is in the section below named Available GAP values.

### Server Properties



### Field definitions

##### Server name

This name can be anything but name the server so that is clear to which instance the data belongs.

##### Data Source

Specify the kind of data source you are using to extract the data. Select “Microsoft SQL” to load the data from a MSSQL database.

##### User Name / Password

This is either SQL authentication or Windows authentication. For windows authentication, use the format of Domain\Username.

##### Database Authentication

Select either SQL authentication or Windows authentication.

##### SQL Server, Instance, Database

Indicate where the data source database is located.

##### SQL Query String

Enter the SQL string that will be run to pull backup job data. Note, SQL comments break the query on some ODBC drivers. Check the list of available data fields that can be imported is in the section below named Available GAP values. For more information review the examples included in this documentation*.*

##### SQL Query Timeout

Indicate the timeout for SQL Query.

##### Date Format

Select the date format. Either “/” or “-“ can be used for the date. So, you can use DD-MM-YYYY and DD/MM/YYYY, and both formats are taken as correct if you select the option “DMY”. Likewise, hours are allowed both on a 24-hour basis and a 12-hour basis.

There are three options to indicate the date format:

* YMD: indicates that the date format is YYYY-MM-DD or YYYY/MM/DD, e.g., 2022-10-18 12:07:28.
* MDY: indicates that the date format is MM-DD-YYYY or MM/DD/YYYY, e.g., 10-18-2022 16:45:26.
* DMY: indicates that the date format is DD-MM-YYYY or DD/MM/YYYY, e.g., 2022-10-18 04:45:26 PM.

##### Number Format

Set the number format according to the format of the numbers that the data to load has.

##### Time Zone

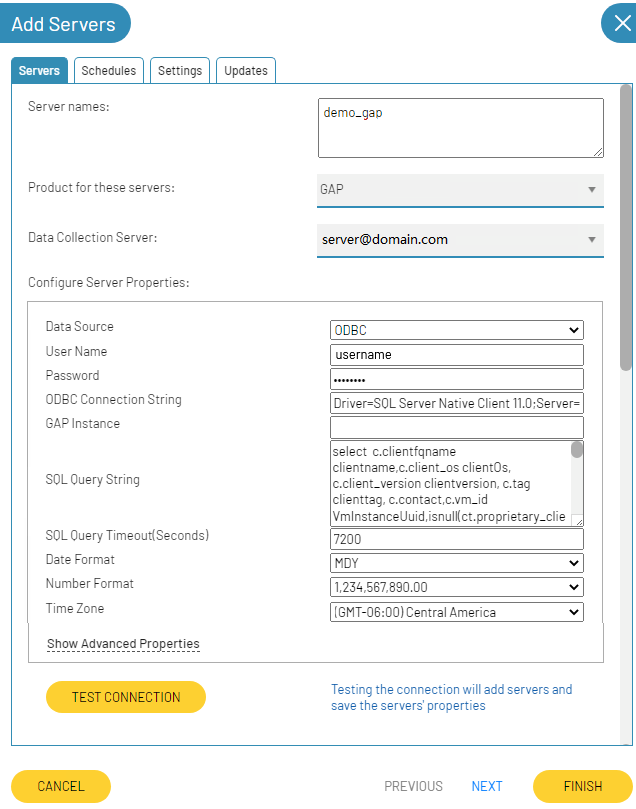
Select the time zone where the server resides. This setting ensures times are displayed consistently in environments that span multiple time zones.

# ODBC

Configure the GAP plugin to work with any ODBC driver. To use a custom ODBC driver, you must install your desired ODBC driver and determine the ODBC connection string required to connect to that driver.

The list of available data fields that can be imported is in the section below named Available GAP values.

### Server Properties



### Field Definitions

##### Server name

This name can be anything but name the server so that is clear to which instance the data belongs.

##### Data Source

Specify the kind of data source you are using to extract the data. Select “ODBC” to load the data from a Custom ODBC.

##### User Name / Password

These are the username and password required to connect to the ODBC driver you configure. The values here are optional, but so that password is encrypted, use the templates in the connection string, e.g. [gap\_odbc\_username\_bocada], [gap\_odbc\_password\_bocada] will get replaced with the values you type

here.

##### ODBC Connection String with Templates

This string is required to connect to the ODBC driver of your choice with some fields optionally using

templates for replacement. Each ODBC driver will fully document the options for the connection string

which vary greatly from one product to another.

|  |  |
| --- | --- |
| Template | Description |
| [gap\_odbc\_username\_bocada] | Replaced during update with Username server property value. |
| [gap\_odbc\_password\_bocada] | Replaced during update with Password server property value |
| [gap\_odbc\_timeout\_bocada] | Replaced during update with SQL Query Timeout server property value |
| [gap\_odbc\_instance\_bocada] | Replaced during update with CMDB Instance server property value |

##### GAP Instance

Indicate the instance that will be used. This field is not required, use it together with the [gap\_odbc\_instance\_bocada] template.

##### SQL Query String

Enter the SQL string that will be run to pull backup job data. Note, SQL comments break the query on some ODBC drivers. Check the list of available data fields that can be imported is in the section below named Available GAP values. For more information review the examples included in this documentation*.*

##### SQL Query Timeout

Indicate the timeout for SQL Query, use it together with the [gap\_odbc\_timeout\_bocada] template.

##### Date Format

Select the date format. Either “/” or “-“ can be used for the date. So, you can use DD-MM-YYYY and DD/MM/YYYY, and both formats are taken as correct if you select the option “DMY”. Likewise, hours are allowed both on a 24-hour basis and a 12-hour basis.

There are three options to indicate the date format:

* YMD: indicates that the date format is YYYY-MM-DD or YYYY/MM/DD, e.g., 2022-10-18 12:07:28.
* MDY: indicates that the date format is MM-DD-YYYY or MM/DD/YYYY, e.g., 10-18-2022 16:45:26.
* DMY: indicates that the date format is DD-MM-YYYY or DD/MM/YYYY, e.g., 2022-10-18 04:45:26 PM.

##### Culture Information

Specify the culture information that will be used when loading the number values.

##### Time Zone

Select the time zone where the server resides. This setting ensures times are displayed consistently in environments that span multiple time zones.

# Available GAP values

When deliver the data to be loaded, all the data required must be provided.

This GAP plugin supports the legacy GAP. We first indicate the fields that the current plugin supports:

## Current Plugin

* The ***csv*** format for the file must contain headers.
* The name of the headers must be as shown in the grid below (case insensitive).
* The columns can be in any order, the header defines the order.
* All backup job data should be provided in one row.
* If any column is repeated, the last value will be used.
* This feature supports no Job Messages or several of them, these messages should be found at the end of the line. More details about this topic below to the grid.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Required** | **Description** |
| ClientName | text | Yes | Client name or VM name typically as shown by the backup product |
| Description | text | No | Unique description, helps ensure the backup can always be unique. |
| CanonicalLevel | text | Yes | Bocada's name for the backup level, e.g., Full, Incremental, Partial, etc. |
| StartDatetime | datetime | Yes | The time the backup started running |
| GroupDate | datetime | No | Start time of the backup group |
| GroupName | text | Yes | A backup group is a set of backups that are run together |
| TargetName | text | Yes | The target is what is getting backed up, e.g., c:\files\... |
| Key | text | No | During managed plugins commit, uniquely keys backup data |
| ByteCount | number | Yes | Set to –1 for "not available", otherwise a 64-bit byte count.  This byte count will typically be equal to BytesBackedUp |
| BytesBackedUp | number | No | Number of bytes backed up by this backup before dedup or compression |
| BytesProtected | number | No | Total number of bytes protected on disk |
| BytesCompressed | number | No | Bytes post compression (by the backup product) |
| BytesPostDedup | number | No | Bytes post dedup |
| ExpirationDate | datetime | No | Date when media data is set to expire, in the local time of the backup server. The backup retention / expiration is an essential piece of data for most customers. |
| RetentionDays | number | No | The number of days this backup data copy will be retained. Used to calculate expirationdatetime if expirationdate is not provided |
| FileCount | number | No | Number of files backed up |
| JobDurationSeconds | number | No | Time in seconds that the backup ran from start to complete |
| GroupDuration | number | No | Number of seconds that backup group was active |
| ProprietaryType | text | No | Normalized Backup destination type, as defined by the backup product e.g. disk or data domain |
| PolicyName | text | No | Name of the backup policy on the backup product that is used to run this backup |
| ScheduleName | text | No | Name of the schedule on the backup product that ran this backup |
| Comment | text | No | Only used for Legacy Gap to define the GroupSession |
| ErrorSet | number | No | This is set dynamically by the framework based on job messages. In Legacy is used to set the field directly; 0 means no errors, and the full Bocada errorset is supported. |
| GroupId | number | No | Not used, can be a reference if needed |
| GroupSession | text | No | Optional, for some products this is useful, e.g., Data Protector, use if needed |
| JobSession | text | No | Can be used if needed to group data during data collection |
| ClientOs | text | No | Backup client operating system |
| ClientVersion | text | No | Backup client version |
| Contact | text | No | Needed for zoning by contact, tenants, regions |
| CanonicalClientType | text | No | Indicates whether a client is an Endpoint or workstation for endpoint products.  Bocada's canonical name for this e.g., Endpoint. |
| ProprietaryClientType | text | No | Backup product's proprietary name for Endpoint e.g., "Workstation" |
| ClientTag | text | No | Used for various purposes |
| VmInstanceUuid | text | No | Only relevant for VM backups.  If the backup is a VM and this data point is available, this should be set |
| VMName | text | No | Name of the virtual machine |

### Job Messages

The job messages should be at the end of the line for better understanding. Zero or multiple messages are allowed.

The next grid shows the fields available to be load:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Required** | **Description** |
| LogMessage | text | Yes | Text message |
| LogErrorCode | text | No | Proprietary error code |
| LogDatetime | datetime | No | The time of the job message |
| IsError | number (0,1) | No | Allow to indicate that a specific message represents an error. 0: it is not an error, 1: it is an error. |

A message can be preceded by another message if there is a repeated column, in this way the application will interpret that there is a new message, and it will be added as such.

The following is an example of how new messages are interpreted:

Text

Description automatically generated

Here we have two different backup jobs, the first backup job (line 1) has three job messages and the second has two messages, let’s see the detail:

Backup Job 1:

Message 1: Marked in yellow, this message has all the fields. The text does not show much information about whether it is an error or not, but it is indicated that it is an error by setting the *IsError* field to 1, and the date and code of the error are also indicated.

Message 2: marked in green, this message only has *logmessage* and *logdatetime*. The plugin will recognize this as a new job message because we already have a column named *logmessage*. The text of the message contains some specific words that allow the process detects an error, even if the column *IsError* is not specified. The end of this message is when is detected a new column that this message has already processed.

Message 3: marked in blue, this message is composed of a single column. The process will try to detect an error message, as the column *IsError* is not defined, the text not always could define a message as an error. Also, as we don’t have the datetime for this message, the *StartDatetime* field will be used.

Backup Job 2:

Message 1: Marked in yellow, this message has all the fields, but not all data is defined in this message. The text hasn’t words indicating an error and *IsError* is set to 0. There isn’t any problem with the null value in the error code field.

Message 2: marked in green, this message only has *logmessage*. The plugin will try to recognize words as an error, but in this case, there are no errors here, so this is only a message and will be stored that way, as logdatetime is null, the *StartDatetime* of the backup job will be used.

Message 3: there are no data for these columns, so they will not be processed.

## Legacy GAP Plugin

The ***csv*** format for the legacy plugin does not contain headers. Therefore, the columns must always be in the same order. The fields defined for this format are shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| ClientName | text | Yes | Client name or VM name typically as shown by the backup product |
| TargetName | text | Yes | The target is what is getting backed up, e.g. c:\files\... |
| StartDatetime | datetime | Yes | The time the backup started running |
| ExpirationDate | datetime | No | Date (time optional) when media data is set to expire, in the local time of the backup server. The backup retention/expiration is an essential piece of data for most customers. |
| HoursGMToffset | number | No | This field will be allowed only to support the Legacy format, but it will be ignored. The Time Zone server properties will used instead. |
| RetentionDays | number | No | The number of days this backup data copy will be retained. Used to calculate *expirationdatetime* if *expirationdate* is not provided |
| ByteCount | number | Yes | Set to –1 for "not available", otherwise a 64-bit byte count.  This byte count will typically be equal to *BytesBackedUp* |
| CanonicalLevel | text | Yes | Bocada's name for the backup level, e.g., Full, Incremental, Partial, etc. |
| GroupName | text | Yes | A backup group is a set of backups that are run together |
| JobDurationSeconds | number | No | Time in seconds that the backup ran from start to complete |
| ErrorSet | number | No | This is set dynamically by the framework based on job messages. In Legacy is used to set the field directly; 0 means no errors, and the full Bocada *errorset* is supported. |
| Comment | text | No | Only used for Legacy Gap to define the *GroupSession* |

# Examples

## CSV File

It does not matter if the file is accessed in a local folder, a network folder or a virtual (http) directory, the file must be a file with a csv extension. It is important that at least the required columns are included.

There are three server properties that have to do with the file format:

1. The file can be delimited by commas, semicolons or pipe, according to this, it must be defined in the server properties so that the data collector mines the information correctly.
2. Another important issue to consider is the date format, you must validate that the date format defined in the server properties is being used in the ***csv*** file. See more details [here](#_Date_Format).
3. Multiple numeric formats are supported, so if numeric values ​​have thousand separators, you must specify the correct separator type. None of the expected values ​​contain decimals, so it is important to consider this when creating the file.

You should consider that if you use a comma delimiter, and the numeric values ​​in your file also use the comma sign as the thousand separators, it is likely that an error will be generated when processing the file. It is recommended that if numeric values ​​contain thousands separator, use another delimiter for the file.

Sample 1:

Graphical user interface, text, application

Description automatically generated

Data:

clientname,clientOs,clientversion,clienttag,contact,VmInstanceUuid,ProprietaryClientType,description,targetname,startdatetime,ExpirationDate,bytecount,jobdurationseconds,errorset,canonicallevel,groupdate,groupname,bytecount,bytesBackedUp,bytesProtected,bytesCompressed,bytespostdedup,filecount,groupduration,ProprietaryType,policyname,schedulename,groupsession,vminstanceuuid,logmessage,logdatetime,iserror,logmessage,logerrorcode,iserror

XAtl-nbu-ws01.testlab.com,,,,,,Other,This is the description,CATALOG\_DRIVEN\_BACKUP,10/31/2022 19:30:00,10/28/2022 15:30:00,-2,0,0,Incremental,10/25/2022 15:30:00,S-Catalog-Policy-2-AdvDisk,-2,,,,,0,0,Unknow,S-Catalog-Policy-2-AdvDisk,-,50506,,Error loading data,10/31/2022 19:30:02,1,,,0

XAtl-nbu-ws01.testlab.com,,,,,,Disk,This is the description,CATALOG\_DRIVEN\_BACKUP,10/31/2022 20:30:00,11/30/2022,-1,0,0,Incremental,11/25/2022,S-Catalog-Policy-2-AdvDisk,-1,,,,,0,0,Unknow,S-Catalog-Policy-2-AdvDisk,Differential-Inc,50506,,Error loading data,10/31/2022 20:30:01,1,,,0

XAtl-nbu-ws01.testlab.com,,,,,,Other,This is the description123456789012345678901234567890123456789012345678901234567890,C:\Program Files\Veritas\NetBackupDB\staging\DARS\_DATA.db,10/30/2022 15:30:05,10/28/2022 15:30:00,408036352,0,0,Incremental,10/25/2022 15:30:00,S-Catalog-Policy-2-AdvDisk,408036352,,,,,18,0,Unknow,S-Catalog-Policy-2-AdvDisk,Differential-Inc,50506,,,,0,This a message and everything went well,C123456789012345,0

XAtl-nbu-ws01.testlab.com,,,,,,Endpoint,Another description,CATALOG\_DRIVEN\_BACKUP,10/31/2022 15:30:00,10/28/2022 15:30:01,12244992,0,0,Incremental,10/25/2022 15:30:00,S-Catalog-Policy-2-AdvDisk,12244992,,,,,706,0,Unknow,S-Catalog-Policy-2-AdvDisk,Differential-Inc,50506,,,,0,,,0

XAtl-ave-wc03.testlab.com,,,,Michael Bolton,,Other,This is the description,/edrv-longrunning-bk2ave02-dataset,10/31/2022 16:00:00,10/26/2022 16:00:00,0,0,0,Incremental,10/25/2022 16:00:00,/IT\_Programming/S-ITP-LongRunning-bk2ave02\_3762\S-ITP-LongRunning-bk2ave02,0,0,19237791500,,,1110010,0,Unknow,/IT\_Programming/S-ITP-LongRunning-bk2ave02\_3762,Unavailable,0,,,,0,,,0

## SQL Query

For the data extraction query, it is important that at least the required columns are included.

The names of the columns must be as they are defined in the [Available GAP values](#_Available_GAP_values) section, this is because the columns can be in any order and are mapped according to the name (case insensitive), for the only columns that the order is considered, it is when loading multiple job messages.

Remember that if you have any column defined more than once (except for job messages fields), the process will only consider the last value found.

It is important to consider the format of the date and the numbers, in case the query requires some type of conversion to comply with the established properties.



Query:

The following query is an example of a query that works for an MSSQL. In case of using an ODBC it depends a lot on what particular driver allows.

|  |
| --- |
| Select c.clientfqname clientname,  c.client\_os clientOs,  c.client\_version clientversion,  c.tag clienttag,  c.contact,  c.vm\_id VmInstanceUuid,  isnull(ct.proprietary\_client\_type,'Other') ProprietaryClientType,  'This is the description'description,  t.targetname,  convert(varchar,l.jobdatetime,120) startdatetime,  convert(varchar,isnull(l.expirationdatetime, l.jobdatetime +6),120) ExpirationDate,  l.jobdurationseconds, l.errorset,  le.canonicallevelname canonicallevel,  convert(varchar,l.groupdatetime,120) groupdate,  g.groupname,  l.bytecount,  l.bytes\_backed\_up bytesBackedUp,  l.bytes\_protected bytesProtected,  l.bytes\_compressed bytesCompressed,  l.bytes\_postdedup bytespostdedup,  l.filecount,  l.groupdurationseconds groupduration,  'Unknow' ProprietaryType,  po.backuppolicy\_name policyname,  po.schedule\_name schedulename,  l.groupsessionname groupsession,  vm.vm\_instance\_uuid vminstanceuuid,  replace(isnull((select top 1 messagetext from jobmessagelog jm  where backup\_id = l.backup\_id), ''),',','') logmessage,  convert(varchar,l.jobdatetime,120) logdatetime,  replace(jm2.messagetext,',','') logmessage,  jm2.proprietaryerrorcode logerrorcode ,  case when jm2.messagetext like '%FAILED%' then 1 else 0 end iserror  from backuplog l  inner join backuptargets t on l.target\_id = t.target\_id  inner join clients c on l.client\_id = c.client\_id  left outer join client\_types ct on c.client\_type\_id = ct.client\_type\_id  left outer join levels le on l.level\_id = le.level\_id  left outer join backupgroups g on l.backupgroup\_id = g.backupgroup\_id  left outer join backuppolicy po on l.backuppolicy\_id = po.backuppolicy\_id  left outer join vmachine vm on c.vm\_id = vm.vm\_id  left outer join jobmessagelog jm2 on l.backup\_id = jm2.backup\_id and jm2.messageordinal = 2  left outer join jobmessagelog jm3 on l.backup\_id = jm3.backup\_id and jm3.messageordinal = 9  where jobdatetime < GETDATE() and jobdatetime > GETDATE()-2 |

# Troubleshooting

* File does not exist:

Check the file name and the access to it. If permission is required, you should set the credentials in the server properties.

* Failed extracting data. Indicated delimiter not found in file (;), please check the file and the server properties:

This error can occur when using CSV files, what should be done is to check that the file delimiter is correctly defined in the server properties.

* Column “AnyColumn”: Input string was not in a correct format:

This error occurs when trying to convert a value that should be a numeric or date value, but the conversion failed. Check the specific line and column and the given value, the date and number formats set in the server properties must be taken into consideration.

* Column name must be defined:

This error means that there is a column with empty name, check the file header.

* No header defined for value “SomeValue”*:*

In the indicated row there is one more value of the columns defined in the header.

* An error occurred executing the statement, please check the query syntax:

Check that the SQL String is correct.

* Error: CSV via HTTP: Failed extracting data. CSV via HTTP: Failed accessing file. An error occurred while sending the request:

If you are accessing an HTTPS URL that requires security, you need to check the credentials you provided in server properties.

# Reporting Notes

All standard Bocada backup jobs reports will work with data collected by the GAP plugin.

# Appendix A - Upload legacy plugin historical data to a new backup server

It is not required or necessary to move the data from the legacy plugin server to the new one, but if the client wishes to have all the information on a single backup server in order to better visualize the history, it can be done.

To achieve this, follow the next steps:

1. Configure the new backup server to connect to the Bocada database using “*Microsoft SQL*” as data source. More details in [here](#_Server_Properties). Set the Date Format as “*MDY*”.
2. Use the following script in the server configuration. Please replace the indicated variables as you need it:

|  |
| --- |
| select c.clientfqname clientname,  t.targetname,  l.jobdatetime startdatetime,  l.expirationdatetime ExpirationDate,  0 hoursGMToffset,  0 retentiondays,  l.bytecount,  le.canonicallevelname canonicallevel,  g.groupname,  l.jobdurationseconds ,  l.errorset,  l.groupsessionname GroupSession,  jm.proprietaryerrorcode LogErrorCode,  jm.messagetext LogMessage,  jm.messagedatetime LogDatetime  from backuplog l  inner join backuptargets t on l.target\_id = t.target\_id  inner join clients c on l.client\_id = c.client\_id  left outer join client\_types ct on c.client\_type\_id = ct.client\_type\_id  left outer join levels le on l.level\_id = le.level\_id  left outer join backupgroups g on l.backupgroup\_id = g.backupgroup\_id  left outer join backuppolicy po on l.backuppolicy\_id = po.backuppolicy\_id  left outer join jobmessagelog jm on l.backup\_id = jm.backup\_id  where l.server\_id = ***@Server\_Id*** |

In the above script, you need to replace the *@Server\_Id* variable with the id of the server you want to load data from, it should be the *server\_id* of the legacy plugin.

If you want to copy only a small set of data, you can add whatever filter conditions you want using Transact-SQL.

1. Run the Data Collation for the new backup server.
2. Change the backup server settings based on the data source you intend to use for data extraction.

# Appendix B - Example to HTTPS Configuration for Testing

This section describes how to set up an HTTPS web site to host a Comma Separated file for the GAP Plugin. You may want to do this to test the plugin.

To test the extraction of data from a CSV file via HTTP or HTTPS, we must have a virtual directory configured. We will configure a virtual directory in IIS for testing purposes with a self-signed certificate, but you can use your virtual directory.

If you don’t have an HTTPS, please follow the next steps:

1. Go to the IIS and in the root of the server and double click on Server Certificates

Text

Description automatically generated with low confidence

1. Create the Certificate Request:
   1. In the right panel, select “Create Certificate Request”, and fill in the fields according to your company.

Graphical user interface, application, email

Description automatically generated

* 1. Click Next.

Graphical user interface, application

Description automatically generated

* 1. Click Next. Specify a file name for the request:

Graphical user interface, text, application

Description automatically generated

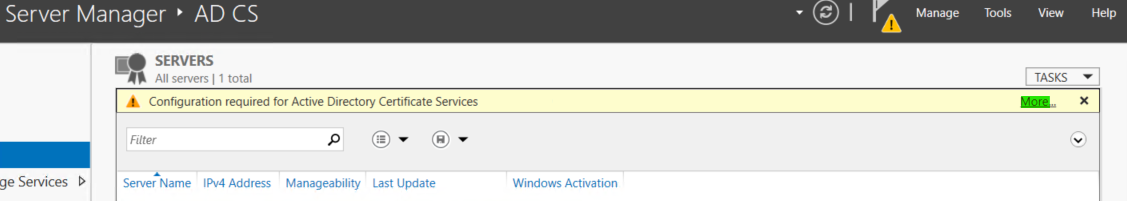
* 1. Click Finish.

1. Configure the “Active Directory Certificate Services”:
   1. Go to the Server Manager and open the “Active Directory Certificate Services”:

Graphical user interface, application

Description automatically generated

* 1. Click on the link “More..”

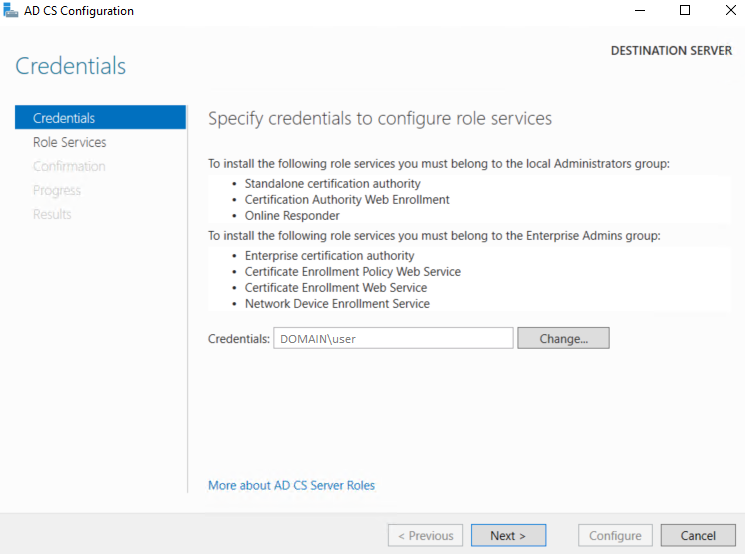


* 1. Click on the link “Configure Active Directory Certificate Services on the destination server”:

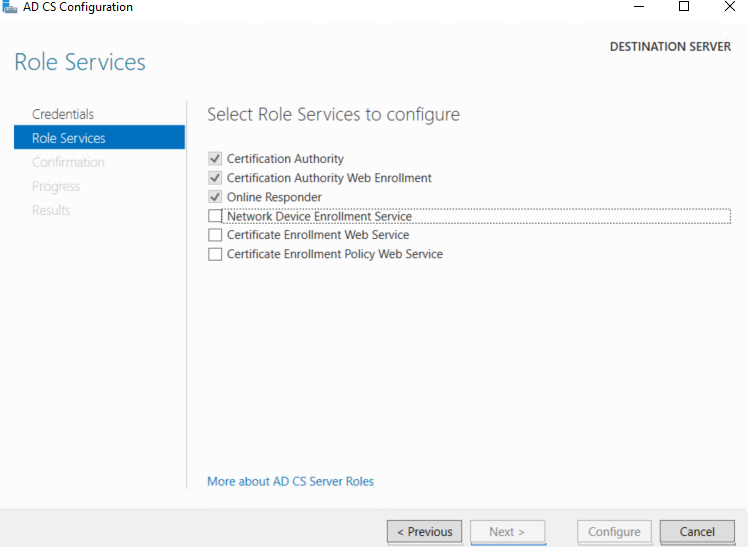
Graphical user interface, text, application

Description automatically generated

* 1. Specify the credentials and click Next:



* 1. Select the checkbox according to the image and click Next:



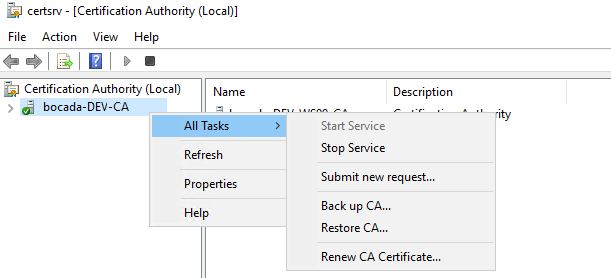
* 1. Select CA independent:

Graphical user interface, text, application, Word, email

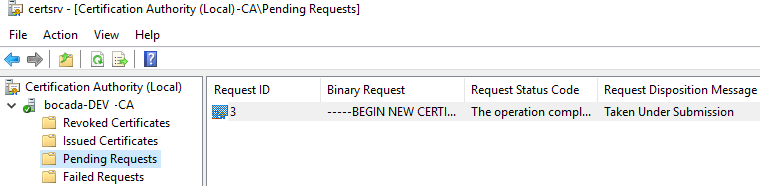
Description automatically generated

* 1. Click next until the configuration is completed.

1. Submit the Certificate Request:
   1. Go to “Certification Authority” and submit a new request: select the file we create in step 2. c.



* 1. Go to “Pending Requests:

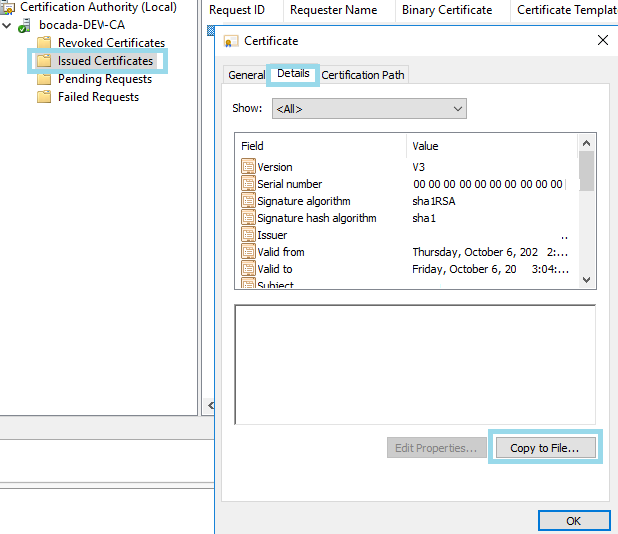


* 1. Right-click on the pending request and click on Issue:

Graphical user interface, application, table

Description automatically generated

* 1. Now, go to “Issued Certificates” and open the certificate. In the tab “Details” select Copy to File:



* 1. Export the certificate as the default configuration.

1. Get back to the IIS configuration, double click in Server Certificates, and in the right panel, click the link “Complete Certificate Request”:
   1. Select the certificate file exported in step 4.d and click OK.

Graphical user interface, text, application, email

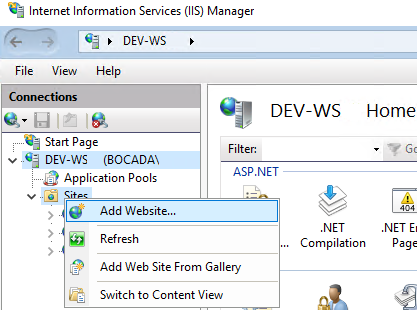
Description automatically generated

1. Create the site:
   1. Create a directory where you are going to have your safe site. You can create one in C:\MySite:

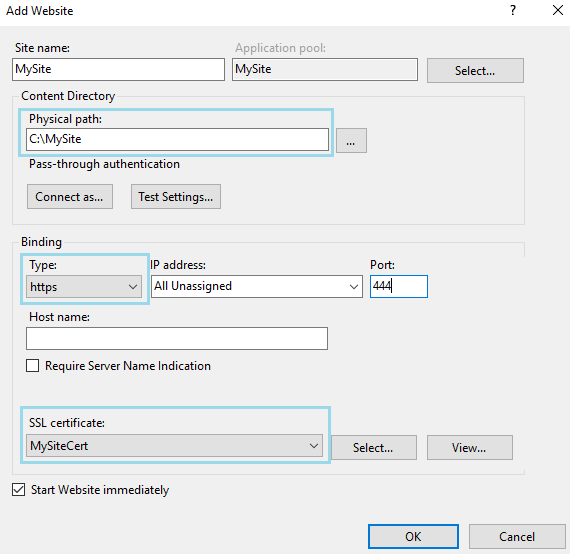
Graphical user interface, application

Description automatically generated with medium confidence

* 1. In the IIS server create the site. In the root right-click and “Add Website”



* 1. Fill in the fields required:
     1. Specify the site name as you like.
     2. In the physical path, select the directory created in step 6. a.
     3. In Binding, change the type to HTTPS
     4. Select the certificate created in step 5. a.

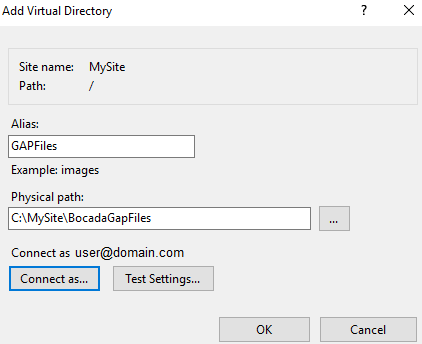


1. Create a Virtual Directory:
   1. Right-click on “*MySite*” o the site created in step 6. c and select the option Add Virtual Directory:

Graphical user interface, text, application

Description automatically generated

* 1. Specify the alias and the physical path, you must also indicate “Connect as” credentials:



If you Test the Settings, it should show that the authentication and authorization are completed.

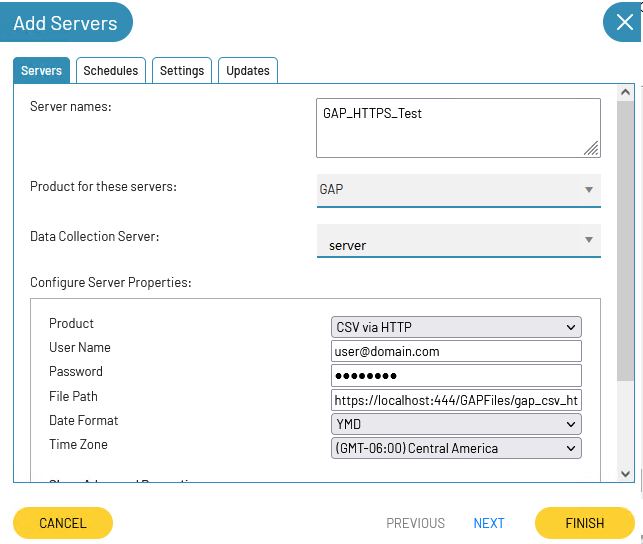
* 1. Select the virtual directory and double-click on “Directory Browsing”

A picture containing diagram

Description automatically generated

* 1. In the right panel, click on the link “Enable”.
  2. Test in the browser the virtual directory.

1. Now we can create or put the CSV files in the virtual directory to be read by the Gap Plugin.
2. Configure the server properties according to your configuration:



1. The username and password must be the ones that you use to configure the virtual directory.
2. In the file path use the URL to access the file via HTTPS.
3. Select the Date Format according to the data.
4. Specify the Time Zone of the backup server.
5. Test the connection of the server properties.

# Technical Support

Please contact us for technical support or a copy of our standard support agreement.

**E-mail:** [support@bocada.com](mailto:support@bocada.com)

**Support Portal:** <http://www.bocada.com/support/>

**Phone:** 425-818-4480

**Copyright © 2022 Bocada LLC.** All Rights Reserved. Bocada and BackupReport are registered trademarks of Bocada LLC. Vision, Prism, vpConnect, and the Bocada logo are trademarks of Bocada LLC. Other product names mentioned herein may be trademarks or registered trademarks of their respective companies.

Protected by U.S patents 6,640,217; 6,708,188; 6,745,210; 7,457,833; 7,469,269; 7,496,614; 8,407,227

The material in this manual is for information only and is subject to change without notice. While efforts have been made to ensure accuracy, Bocada LLC assumes no liability resulting from errors or omissions in this document or the use of the information contained herein.

Bocada LLC reserves the right to make changes in the product design and documentation without reservation and without notification to its users.