VERSION 1.0

DECEMBER 17, 2018

Apps for Greentree

CSV DATA EXTRACTOR

APP NUMBER: 010140

Powered by:

MYOB Greentree

TABLE OF CONTENTS

FEATURES

1. **Provides flexible CSV extract capability.**

   Fully configurable extract to csv, include linked objects oids, collections, returned data from simple methods, tree values, UDFs and plugin properties.

   Full or incremental extracts.

IMPORTANT NOTES

- We recommend that you test the configuration of the App thoroughly in a test system prior to deploying the App in your live Greentree system.

OTHER REQUIREMENTS

Greentree Modules: None.

Associated Apps: None.

USER INSTRUCTIONS

EXTRACT DEFINITION

The folder location set up in app configuration must contain a file to define what is to be extracted.

The layout of this file is a series of lines with each line a Class to be extracted, and required inclusions.

The fields to be extracted must be listed separated by commas, with any qualifiers included.

The format that needs to be used is as follows:

<schema>|<Class>(_<collection>)|AI,<field>,<field>|Q/M/P/T/U/F/D/H

Where:

_collection = class collection eg allTreeZones

A = All instances of the class including sub classes

I = All incremental instances of the class.

On a Verde site all customers are stored at VCustomer subschema level so VCustomer without the |A will find the data, but the class Customer will find nothing unless the A qualifier is used eg Customer|A. In general, it is better to use the Greentree Class with the “A” option, but note that the Transaction Class will return many records if “A” is used. User the Verde Class if Verde attributes are required.

The “I” option will return only instances of the class that have been changed or deleted. Separate *_Changed.csv and *_Deleted.csv files are created if this option is used. The first time this is used all records will be returned in the *_Changed.csv file, thereafter only changed records. Records will be appended if the file already exists.

Field Modifiers:

M = method

P = Plugin Property

O = Object - convert to Oid.

T = Tree branch value

U = UDF value

Modifiers that apply to all properties other than Oids eg you can use UD for a UDF value formatted as a date:

F = TimeStamp - format as yyyy/MM/dd HH:mm:ss

D = Date - format dd/MM/yyyy

H = Time - format HH:mm:ss

B = Boolean – returns 0 or 1 rather than false or true.

Modifiers that apply to all properties:

Q = Quote the value. Note that quotes will also be applied if data requires it. Double quotes within data will be converted to single quotes.

EXAMPLE DEFINITIONS WITH EXPLANATIONS AND SAMPLE OUTPUT

1. GreenTreeSchema|Customer|A,oid,code|Q,name|Q,myARControl

All instances of Customer class, listing

- oid
- code – quoted
- name – quoted
- myARControl.oid

2. GreenTreeSchema|POShipment|A,oid,myPOControl,myCurrency,reference|Q,description|Q,documentDate|D,currencyRate,arrivalDate|D,enteredBy,entryTimeStamp|F

All instances of POShipment class, listing

- oid
- myPOControl.oid
- myCurrency.oid
- reference – quoted
- description – quoted
- documentDate – format dd/MM/yyyy
- currencyRate
- arrivalDate - format dd/MM/yyyy
- enteredBy
- entryTimeStamp - format as yyyy/MM/dd HH:mm:ss

3. GreenTreeSchema|Customer_allTreeZones|A,customer_oid,treeZone_oid

All instances of customers tree zones, listing

- customer.oid
- treezone.oid

  Note for a collection the output file will be named by the main class and collection so, for example Customer_allTreeZones.csv

4. GreenTreeSchema|SOPackingSlip|A,oid,mySOControl,myBranch,myCustomer,myLocation,reference,documentDate|D,fcNetAmount,gti_GetStatusName|M

All instances of SOPackingSlip listing

- oid
- mySOControl.oid
- mBranch.oid
- myCustomer.oid
- myLocation.oid
- reference
- documentDate - format dd/MM/yyyy
- fcNetAmount
- status name returned from method gti_GetStatusName

5. GreenTreeSchema|INLocation|A,oid,myWarehouse,findMyWarehouse|MO,fullPath|Q

All instances of INLocation listing

- oid
- myWarehouse.oid
- warehouse returned from findMyWarehouse method as oid
- fullPath – quoted

6. GreenTreeSchema|StockItem|A,oid, myINControl,myAnalysisCode,code,description|Q,Contract|T,Manager|UQ

All instance of StockItem listing

- oid
- myINControl.oid
- myINAnalysisCode.oid
- code
- description – quoted
- contract (tree value)
- manager (UDF value)

Sample Output:

oid,myINControl,myINAnalysisCode,code,description,Contract,Manager

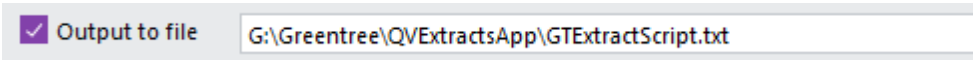
2576.76,2754.1,2610.13,MSOFTWHEELOPTMOUSE,"Microsoft Optical Mouse",Contract 123,"Bryce Jones"

For further examples of Classes and common field used, please contact your Greentree Consultant

EXTRACT EXECUTION

Apps4GT General – CSV Data Extractor is the system script made available by the App.

The Script will look for the Export Template.txt file in the directory specified in the APP Configuration. However, when running and scheduling the script an output file can be specified, eg



The directory specified here will be used instead of the App control directory. This allows a different directory and template to be used for Incremental extract files (using the "I" option) and Full extract files (without using the "I" option).

If any errors are encountered in the Export process, they will be written to Export Errors.log, in the same location.

When the script is run, the extractions will be written as individual csv files, in the same folder location, with files named according to the class extracted. In addition an "Export Check.txt" file is generated containing a line for each class extracted with a count of lines. If the extracts are to be loaded into another system this can be used to check that the load is complete.

The header line for each csv file contains column headings, being the field names in the template.

Non-quoted text fields (oid, date, timestamp and numeric fields are ignored) the system will check for quotes and commas. If any are found then " (double quotes) will be changed to ' (single quotes). The presence of a comma will force the field to be quoted.

Where a myXxx reference, the oid will be used.

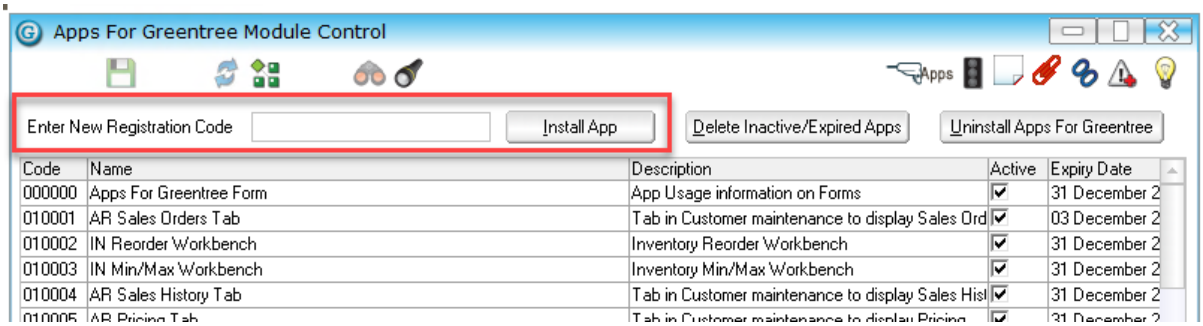
Use of methods is limited to where a single returned value.

IMPLEMENTATION GUIDE

Please refer to the Important Notes section above before installing and configuring this App

APP INSTALLATION

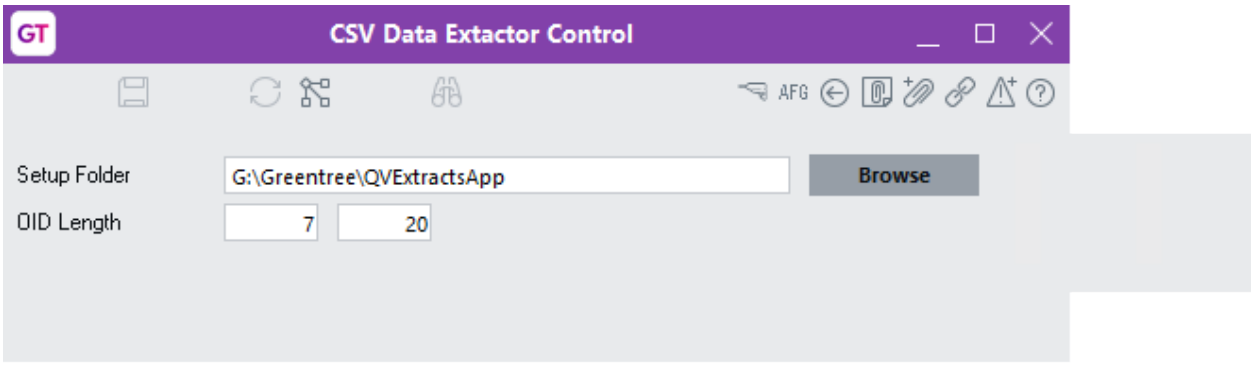
1. Log into Greentree as the Super user
2. Select the menu item | System | Apps For Greentree | Apps Module Control |
3. Enter the New Registration Codes supplied and click Install App



4. Select/Highlight the CSV Data Extractor App.
5. Click on the Edit Users button and select the users who will be configured to use this App, for which companies.
6. Save and Close the form.

APP CONFIGURATION

1. Select/Highlight the CSV Data Extractor App.
2. Click on the Change button.



Setup Folder	Browse to select folder for holding file for extraction definitions (and for extracted output).
OID Length	OIDs are 7.20 decimal numbers. Some applications cannot handle decimals this large. The separator can be changed to another character, eg "_" making the OID a Text field. In the

JadeGT.ini file specify the separator to use, eg:
[JadeOdbc]
OidFieldSeparator=_

INCREMENTAL EXTRACTS

A Full Extract should be scheduled to run overnight so that performance for users is not affected during the day. This is achieved using an Export Template without the “I” option.

An Incremental Extract is achieved using an Export Template with the “I” option on selected classes. The script should be run with an output file located in the same directory as the Export Template file.

If files need to be moved to another server a Batch command file can be used. With Incremental extracts multiple systems need access to the same files at the same time. To handle this a blocking file is used.

When the BlockingFile with the name “Export Mutex.txt” exists in the location the Greentree Script will not run. The script will create this file and remove it when extracts are completed.

Similarly, the receiving application, eg QV will need to process incremental files before the next set of files arrive and override the previous files. The Batch file needs to wait until the GT script has completed. The last Class in the Export Template should be a nominal class, eg SystemRoot. When it is created then process is finished.

The sample batch command file below:

1. would be scheduled with a Windows scheduled task. A separate batch command file for full and incremental would be used with separate scheduled times. The incremental batch file can run throughout the day.
2. will exit out if the GT script BlockingFIle Export Mutex.txt exists.
3. will exit out if the Receiving application has not removed its blocking file, eg QVBlockingFile
4. will select either a Full or Incremental Export Template file to be used (not necessary if using separate directories when running the script)
5. will delete the csv file created last by the script eg SystemRoot.CSV
6. will loop indefinitely checking every 30 seconds for the last csv file to be created.
7. will create GT script BlockingFIle Export Mutex.txt
8. will MOVE files to another server for a waiting application to process,
9. will remove GT script BlockingFIle Export Mutex.txt once completed the move.
10. Will create the QVBlockingFile to indicate that the receiving application can process the files.

```
@echo off
SET LookForFile="G:\Greentree\QVExtractsApp\SystemRoot.csv"
SET BlockingFile="G:\Greentree\QVExtractsApp\Export Mutex.txt"
SET QVBlockingFile="\\192.168.100.36\d$\Data\Greentree\QVExtracts\Import Mutex.txt"

rem Copy CSV extracts for QV Development purposes
rem This script should run overnight.

rem If mutext file exists do nothing, GT is creating files or QV is busy processing.
IF EXIST %BlockingFile% GOTO Exit
IF EXIST %QVBlockingFile% GOTO Exit

 rem set Emport Termplate to Full
 COPY /Y "G:\Greentree\QVExtractsApp\Export Template_Full.txt" "G:\Greentree\QVExtractsApp\Export Template.txt"


 rem delete lookingfor file and wait for the GT extract to recreate it (last file in template)
  IF EXIST %LookForFile% DEL /f %LookForFile%

 rem loop looking for file
 :CheckForFile
  IF EXIST %LookForFile% GOTO FoundIt
  REM Wait 30 seconds and then recheck . Needs to be much longer when not testing
  TIMEOUT /T 30 >nul
  GOTO CheckForFile


 :FoundIt
 echo "T:Test\Copy CSV Extracts.bat - Creating Export Mutex.txt to prevent GT overriding files while taking" >
"G:\Greentree\QVExtractsApp\Export Mutex.txt"
 robocopy "G:\Greentree\QVExtractsApp" "\\192.168.100.36\d$\Data\Greentree\QVExtracts" *.csv
/LOG+:"G:\Greentree\logs\SynchBackup.log" /MOV
 rem set Export Template back to incremental
 COPY /Y "G:\Greentree\QVExtractsApp\Export Template_Incremental.txt" "G:\Greentree\QVExtractsApp\Export Template.txt"
 DEL /f "G:\Greentree\QVExtractsApp\Export Mutex.txt"
 echo "T:Test\Copy CSV Extracts.bat - Creating Import Mutex.txt to prevent files being taken before QV is finished with previous files"
> %QVBlockingFile%
 GOTO Exit


:Exit
exit
```

TROUBLE SHOOTING INCREMENTAL EXTRACTS

Greentree Script is running but not creating any files. Check if the Export Mutex.txt exists and remove it. If the script was run with an output file in a location other than App Control the files may be getting created in another directory. If in doubt, reschedule the script.

Only getting Incremental files. Check if the “I” option is being used in the Export Template

Files are not being copied to another server by the batch command. Check if the Import Mutex.txt exists and remove it