CIMERA PlugIn Library

Dataloader PlugIn

|  |  |
| --- | --- |
| Author | Gwyn Carwardine |
| Date | 3/10/14 |
| Version | 1.1-B |

Contents

[1 Introduction 3](#_Toc400133921)

[1.1 Availability 3](#_Toc400133922)

[1.2 Warranty 3](#_Toc400133923)

[2 Using the Dataloader 3](#_Toc400133924)

[2.1 Accessing the Dataloader 3](#_Toc400133925)

[2.2 Dataloader window 3](#_Toc400133926)

[2.3 Validate 4](#_Toc400133927)

[2.4 Load 4](#_Toc400133928)

[3 CSV file format 5](#_Toc400133929)

[3.1 Users 5](#_Toc400133930)

[3.1.1 Header Row 5](#_Toc400133931)

[3.1.2 Data Row 5](#_Toc400133932)

[3.1.3 Examples 5](#_Toc400133933)

[3.2 Items 6](#_Toc400133934)

[3.2.1 Header Row 6](#_Toc400133935)

[3.2.2 Data Row 8](#_Toc400133936)

[3.2.3 Examples 9](#_Toc400133937)

# Introduction

The Dataloader PlugIn enables quick loading of bulk data into Cimera from comma-separated (CSV) files. It can handle thousands of lines of data.

The Dataloader currently supports loading of Users and Items (but not Item Versions). The design is modular allowing support for additional data types to be easily added in the future.

The Dataloader will be incorporated into Cimera base functionality in a future release.

## Availability

The Dataloader PlugIn is a non-customer specific plugin freely available to all customers.

## Warranty

The Dataloader PlugIn is currently not an officially supported offering and is supplied to customers on an As-Is basis. Errors may be reported by customers and will be fixed as time allows. Dataloader does not form part of your support agreement and no priority incidents may be raised against it.

When the Dataloader becomes part of the base product it will come under standard support.

# Using the Dataloader

## Accessing the Dataloader

The Dataloader may only be executed by users with the Administrator privilege or those that are members of a Cimera group named **#Dataloader** and is accessed via the Shortcut Bar (Data Management section):



Alternatively it may be accessed from the main menu’s Custom entry:



## Dataloader window

The Dataloader opens a new window:



The top text box accepts a file path to the CSV to be loaded. This can be pasted in or the Browse button can be used to locate the appropriate file.

The bottom pane contains log output from the load process. Each row of the CSV is read and written to the log as well as confirmation of loading or any errors generated. The Clear Log button clears the log output pane.

The logging detail produced can be set via the dropdown to one of the following:

* Info – Displays all errors, warnings and information (e.g. success) messages
* Warning – Displays only errors and warnings
* Error – Displays only errors
* Debug – Displays verbose information about how the information is being parsed. Not for the faint-hearted!

The two actions available are Validate and Load:

## Validate

When validating the Dataloader will read the CSV and check that the structure is valid and also that the content is valid to some degree. It does not fully validate the content however:

1. Does not validate that an item of data does not already exist
2. Does not validate that a data item has a valid value. For example if an attribute has an integer format and a non-integer is supplied then this will not be validated. Similarly if an attribute has a range of valid alphanumeric values (i.e. a list) then this will not be validated.

## Load

When loading the Dataloader will first validate the file (as per the Validate function). It will then attempt to create a new data item. If the data item already exists or fails Cimera validation then an appropriate message will be produced and the loading will terminate unless the “Continue to load items after a failure” box is checked.

When a load terminates due to error all items loaded up to that point will have been added to Cimera and therefore rerunning the load will cause errors (item already exists). It is prudent therefore to edit the CSV file to remove those successfully added before rerunning.

# CSV file format

**Warning!!** When saving files from Excel as file type CSV it will often create improperly structured CSV files with extraneous commas on the ends of rows if the header row contains more values than the data rows. For this reason always check, and edit if necessary, any CSV files produced by Excel before using them with the Dataloader.

A CSV file must contain a header row followed by one or more data rows. Any blank lines are ignored. The header row identifies how the following data rows are to be interpreted.

The file may comprise multiple different loads as long as each has a header row followed by data rows. For example a valid file may contain:

Header-row-for-Projects

Project 1 data

Project 2 data

Header-row-for-Environments

Environment 1 data

Environment 2 data

Environment 3 data

The format for the header and data row are specific to the type of data being loaded and are described in the following sections

## Users

### Header Row

The header row must be $$USER

### Data Row

The data row has the following format:

Userid, firstname, surname, email address, administrator?, can user logon?, user readonly?, password [,role|group]…

Those with a ? shown are Boolean values and should be either **true** or **false**.

If Cimera is using Windows Authentication then the password is irrelevant but a value still needs to be supplied (e.g. abc) but will never be used by Cimera.

The square brackets around role|group indicate that it is optional data. The ellipses (…) following indicate that it may be repeated as necessary to give the user multiple group memberships.

Role and Group may be specified using either their internal id (ROL:n and GRP:n) or via their names. References to role and group are case insensitive.

### Examples

$$USER

TimT,Timothy,Taylor,tim.taylor@propelsystems.com,false,true,false,elephant1,member|testing

Jack,Jack,Jones,jack.jones@propelsystems.com,true,true,false,bluemoon,manager|dev,manager|testing

SmithG,Graham,Smith,smithy@propelsystems.com,false,true,false,gray2014,ROL:3|GRP:12

## Items

### Header Row

The header row for items has the following format:

$$ITEM,itemType,attributeOrLink[|attributeOrLink]…[,attributeOrLink=default-value]…

The square brackets indicate an optional clause and the ellipses indicate a repeatable clause.

*Note: all data in the header row is case insensitive.*

#### Item Type

itemType tells the Dataloader what types of Item are following. This can be either the internal id (e.g. TYP:13) or the Item Type’s name (e.g. Project)

#### Attributes / Linked Items

attributeOrLink is a reference to an attribute (field) or to a Valid Link.

An attribute reference may be via the internal id (e.g. FLD:A65) or via the attribute’s name (e.g. Portfolio).

A reference to a Valid Link must be via its Valid Link key (e.g VLK:n:x where x is the direction F-forward / R-reverse / N-non-directional.)

Cimera will ensure that all linked items already exist so that it may relate the new item to these linked items. It may be desirable to continue to load the item even if one or more of the linked items do not exist. This can be achieved by specifying the tilde symbol before the valid link key, e.g. ~VLK:2:R. If this is specified then a warning message will be produced but the item will still be created.

The easiest way to determine the correct Valid Link key is to use the Process Modeller and look at the Form Display Items:



In the example above the Programme is related to Projects via the Valid Link key VLK:11:F (note that the Object ID column is generally far over to the right and the column was moved for this screenshot).

Essentially the attributes / Valid Links represent the columns, and the order of the columns that will follow in the data rows. A sample header row:

$$ITEM,programme,**ciname|portfolio|cidescription|VLK:11:F**,cibucket=projects

This indicates that there are 4 data items to be expected, separated by commas, in each data row.

*Note that the list of attributes / links specified in the header row are separated by a vertical bar | and not by commas.*

#### Default Values

The final entry in the header row is zero or more default attribute / link values that are to be used for each item to be created.

The purpose of these default values is to remove the need to specify identical data on each data row. For example if 100 items are all being added and they will belong to the same bucket then it is inconvenient to have to specify the bucket on every data row.

Instead we can specify cibucket=projects as a default in the header row and this will apply this attribute value to each of the items being created.

The attribute does not have to exist as a column in the data rows. However if it appears in the list of columns and also as a default then the Dataloader will use the value specified in the data row unless it is empty in which case it will use the default value instead.

Multiple default value name=value pairs may be specified separated with commas e.g. cibucket=projects,ciowner=ems,cistatus=exists

Where a value represents a Cimera object (e.g. bucket, group/user, status) either the internal id may be used (e.g. BKT:1) or the object’s name (e.g. Projects). When specifying a user within a group then it should be of the format group/user e.g. GRP:3/USR:27 (using internal ids) or Testing/Timothy Taylor (using names)

#### Automatically generated Item IDs

Some Item Types have their primary id (attribute ciname) automatically generated for them. However the Dataloader still requires you to provide a value for this attribute, even though it won’t be used. The simplest way when loading items with automatically generated ids is to add a default value ciname=notused.

### Data Row

Each data row must have a number of columns equal to the number of attribute / link columns specified in the header row.

#### Attribute formatting

If any column validly contains a comma then the column data must be enclosed within double quotes e.g. “I include, or should I say contain, a couple of commas”.

Any non-text fields must be convertible to the appropriate data type of the attribute. A date can be specified in any of the popular formats (e.g. 1/1/2015, 1 jan 2015, 1-1-2015). Times should be specified using the 24 hour clock. A combined date and time can be specified separated by a space e.g. 1/1/2015 12:34.

Boolean values should be specified as either *true* or *false*.

As with default values in the header row, where a value represents a Cimera object (e.g. bucket, group/user, status) either the internal id may be used (e.g. BKT:1) or the object’s name (e.g. Projects). When specifying a user within a group then it should be of the format group/user e.g. GRP:3/USR:27 (using internal ids) or Testing/Timothy Taylor (using names)

#### Linked Items

Where a column represents a Valid Link the target linked item can be referenced either by its internal id (e.g. CI:193) or by its primary id (ciname attribute – e.g. ServerA)

If a link target can be of multiple types then the name of the target may not be unique across all of those types. For example if a server can be linked to either an application or a system and the name “contax” exists as both an application and a system then the Dataloader will not know which is to be used and will stop with an error message. In this case either the internal id (e.g. CI:193) can be used or the item name can be qualified by prefixing with item-type\ e.g. application\contax

Multiple linked items can be specified by separating the target items with a double vertical bar || e.g. ServerA||ServerB||ServerC or CI:123||CI:456||CI:789

If a link has attributes then these may also be specified within the data by separating with a single vertical bar, syntax is link-target|attribute1=value|attribute2=value etc. As when referencing attributes in the header row either the attribute’s internal id (e.g. FLD:A22) or its name (e.g. DeploymentMethod) can be used.

Multiple linked items with attributes can be specified by combining the two syntaxes above e.g. ServerA|DeploymentMethod=Manual||ServerB|DeploymentMethod=Auto

### Examples

#### No linked items

$$ITEM,project,ciname|ciowner|title|goLiveDate,cibucket=projects

PRJ1,Project Managers/Jason Knight,Contract system replacement,1/1/2015

PRJ2,Project Managers/John Derry,”Tax payments,claims and reconciliation”,31 Dec 2014

PRJ3,Programme Managers/Keith Todd,”SAP billing engine migration”,30-6-2016

#### Single linked items

Note: vlk:4:r represents a link from a server to an environment and there are environments in Cimera named PPT1 and Devenv1-4

$$ITEM,server,ciname|vlk:4:r|platform,cibucket=assets,ciowner=asset mgt

ServerA,PPT1,Unix

ServerB,DEVENV1,Unix

ServerC,PPT1,Windows

#### Single optional linked items

Note: vlk:4:r represents a link from a server to an environment and there are environments in Cimera named PPT1 and Devenv1-4

$$ITEM,server,ciname|~vlk:4:r|platform,cibucket=assets,ciowner=asset mgt

ServerA,PPT1,Unix

ServerB,DEVENV1,Unix

ServerC,PPT1,Windows

#### Multiple linked items

Note: vlk:4:r represents a link from a server to an environment and there are environments in Cimera named PPT1 and Devenv1-4

$$ITEM,server,ciname|vlk:4:r|platform,cibucket=assets,ciowner=asset mgt

ServerA,PPT1,Unix

ServerB,DEVENV1||DEVENV2||DEVENV4,Unix

ServerC,PPT1||DEVENV3,Windows

#### Multiple linked items with attributes on the link

Note: vlk:5:F represents a link from an environment to a person and as part of the link there is an attribute representing role – the role that person has on that environment.

$$ITEM,environment,ciname|vlk:5:f|cistatus,cibucket=Testing,ciowner=Test

PPT1,Steve Moore|role=techie,In Use

DEVENV1,Paul Gittins|role=unix expert||Frank Robbins|role=superuser,In Use

DEVENV2,Sayed Romal|role=unix expert||Frank Robbins|role=superuser,In Use

DEVENV3,Simon Collins|role=Windows cleaner,Out of service

#### Using Cimera internal ids

It is generally easier to use names but it is equally as valid to use Cimera’s internal ids for attributes and objects. Using the previous example:

$$ITEM,typ:3,fld:a$ciname|vlk:5:f|fld:o$cistatus,fld:o$cibucket=bkt:4,fld:o$ciowner=grp:52

PPT1,CI:74|role=techie,sta:13

DEVENV1,CI:94|role=unix expert||CI:101|role=superuser,sta:13

DEVENV2,CI:932|role=unix expert||ci:101|role=superuser,sta:13

DEVENV3,ci:830|role=Windows cleaner,sta:17

Names and ids can be mixed and matched as desired. Sometimes there may be conflict where a name has been used more than once (i.e. is not unique) and so id needs to be used to provide a unique reference.

When using ids the id is case insensitive, in other words BKT:4, bkt:4 and BkT:4 are equally acceptable