

# CIMERA INTERFACES

## Overview

Release 4.2.x

Version 1.0, 10-Dec 2015

Gwyn Carwardine, Jon Bentley

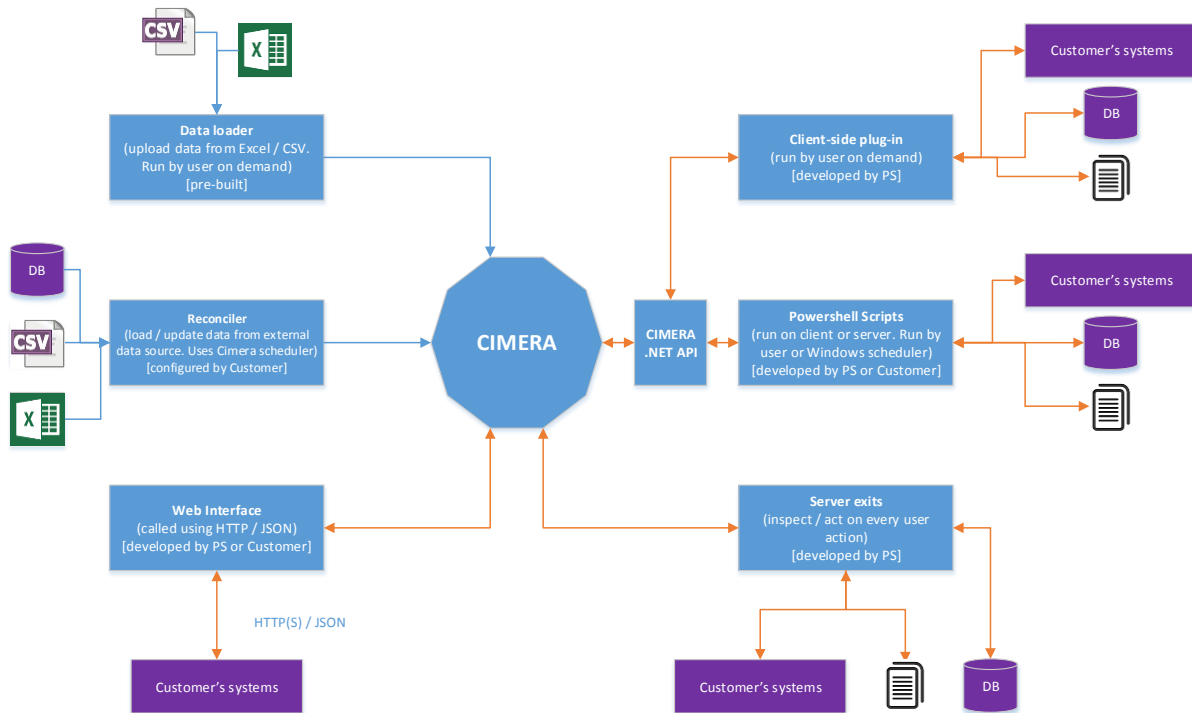
gwyn.carwardine@propelsystems.com  
jon.bentley@propelsystems.com

## Contents

1	Overview.....	3
2	Data Loader .....	3
3	Reconciler .....	3
4	Web Interface .....	3
5	Cimera .NET API.....	4
6	Client-side Plug-in .....	4
6.1	Plug-in Types .....	4
6.1.1	Standard.....	4
6.1.2	Object Action .....	4
6.1.3	Item Display .....	4
6.1.4	Custom Action.....	4
6.1.5	Related Items .....	4
6.1.6	Valid Link Filter .....	4
6.1.7	Pre-Create.....	4
6.1.8	Post-Create .....	5
6.1.9	Pre-Update.....	5
6.1.10	Post-Update.....	5
6.1.11	Pre-Delete.....	5
6.1.12	Post-Delete .....	5
7	Powershell Scripts.....	5
8	Server Exits .....	5
8.1	Exits Points .....	5
8.1.1	Update Server .....	5
8.1.2	Query Server .....	5
8.1.3	Index Query Server .....	6

## 1 Overview

The following diagram shows the variety of interfaces that are available within Cimera.



## 2 Data Loader

This is supplied as a plug-in (run on demand by the user) and allows loading of data from either comma separated (CSV) files or Excel spreadsheets. A simple mapping in the source data tells Cimera how to translate the source data to Cimera items. This is typically used as part of an initial data-migration or for subsequent bulk-loading of new data.

## 3 Reconciler

The Reconciler is built into Cimera and performs similar functionality to the Data Loader, except that the data source and required mapping is configured in Cimera as an “External Data Source”. As well as loading new data it will also compare current Cimera data with the external data source and update the data within Cimera to match. The Reconciler enables Cimera to extract information directly from an external data source. The Reconciler can read data from any OLEDB data source (this includes the majority of commercial database systems as well as CSV and Excel files). If translation of data is required this can be achieved by exporting data from the foreign system, manipulating via scripts and then placing the translated data in the Reconciler’s input directory.

## 4 Web Interface

Cimera exposes its API over HTTP (or HTTPS) using the JSON standard data format. This enables web systems to interface more natively with Cimera but also enables cross-platform integration as the HTTP / JSON standard is platform agnostic.

## 5 Cimera .NET API

The Cimera .NET Advanced Programming Interface (API) allows any .NET-friendly programming language (for example .NET itself or Powershell) to make calls to perform any Cimera function. API users are subject to standard Cimera authentication and authorisation.

## 6 Client-side Plug-in

Cimera functionality may be extended by plug-ins. These are .NET compiled units of code that are stored on the server and automatically downloaded to the client when the user logs in. Plug-ins may be used to provide extra functionality as either standalone functions (the user can simply execute at any time) or they may be integrated into Cimera's context (right-click) menus such that when a user right-clicks on an item, or items, the plug-ins provide extra actions that may be performed against the selected item(s). These plug-ins can do anything – either within Cimera or external to Cimera. Other sorts of plug-ins are also called when the user creates, updates or deletes items which can provide extra validation or automate certain processes. For example when a new Project is created in Cimera there may be other related items that should also be created – a plug-in can be used to provide this automation

### 6.1 Plug-in Types

#### 6.1.1 Standard

A Standard plug-in appears in Cimera's Shortcut Bar and in the Plug-In menu.

#### 6.1.2 Object Action

This plug-in can be used to extend the right-click menu when one or more items are selected.

#### 6.1.3 Item Display

The Item Display plug-in allows a bespoke form to be displayed when creating or updating items of a specified Item Type. This may be used in cases where very precise layout is required or where data needs to be presented in a more compact format than the standard Item Display form.

#### 6.1.4 Custom Action

A Custom Action plug-in can be added to an Item Display form for a specified Item Type. It appears as a button on the form and when the button is pressed the plug-in is executed. The plug-in has access to all fields on the form.

#### 6.1.5 Related Items

When displaying an item it may be desirable to display a list of related items that meet specific criteria. For example when displaying a Project it may be helpful to display a list of other items that are indirectly related to the Project and would not normally be displayed but would be useful for the user to see.

#### 6.1.6 Valid Link Filter

When creating or updating an item and selecting to add linked items it may be desirable to apply additional custom filtering rather than simply displaying the list of all available items.

#### 6.1.7 Pre-Create

Called after the user has selected to create an item but before the client has passed the instruction to the server. Potential uses includes performing additional complex validation or adding additional information to the new item.

### 6.1.8 Post-Create

Called after the server has successfully created an item. Potential uses include triggering an external event or performing more actions within Cimera.

### 6.1.9 Pre-Update

Called after the user has selected to update an item but before the client has passed the instruction to the server. Potential uses includes performing additional complex validation or modifying the information in the updated item.

### 6.1.10 Post-Update

Called after the server has successfully updated an item. Potential uses include triggering an external event or performing more actions within Cimera.

### 6.1.11 Pre-Delete

Called after the user has selected to delete an item but before the client has passed the instruction to the server. Potential uses includes performing additional complex validation.

### 6.1.12 Post-Delete

Called after the server has successfully deleted an item. Potential uses include triggering an external event or performing more actions within Cimera.

## 7 Powershell Scripts

Powershell is a powerful scripting language available on Windows servers and workstations. It is the Windows equivalent of the UNIX “shell” scripting language. Powershell can be used to interact with systems that expose a Command Line Interface (CLI), an API or Web Services. It provides functions to enable comprehensive manipulation of textual data and XML. Most Windows administrators or engineers will be familiar with writing Powershell scripts. Propel Systems provide a library of Cimera Powershell functions to simplify script writing.

## 8 Server Exits

These are effectively server-side plugins. The name “exit” comes from mainframe terminology, there are also sometimes called “triggers”. Whenever the server performs an action any exits are called prior to the action being completed (as might be used for custom complex validation) and after the action is completed (as might be used for sending data or commands to a foreign system). For performance reasons these are written in .NET and are written for the customer by Propel Systems.

### 8.1 Exits Points

#### 8.1.1 Update Server

The Update Server processes all requests to make changes to Cimera data or Cimera’s metadata. The exit supports three functions: BeforeProcessingRequests, AfterProcessingRequests and AfterProcessingCommit. This exit may be used to validate or modify data, to export data or to trigger external events.

#### 8.1.2 Query Server

The Query Server processes all read-only activities; those that retrieve data. The exit supports two functions: BeforeProcessingRequests and AfterProcessingRequests. This exit may be used to inspect and/or modify a user’s request, to modify data prior to returning to the user or to add supplemental data retrieved from an external source.

### 8.1.3 Index Query Server

The Index Query Server process all full-text search requests from the user. The exit supports two functions: BeforeProcessingRequests and AfterProcessingRequests. This exit may be used to inspect and/or modify a user's request, to modify data prior to returning to the user or to add supplemental data retrieved from an external source.