



Service Description - Pacemaker Community  
Edition (CE) v1.3.0

# Technische modul informationen

<b>Hersteller</b>	TechDivision GmbH
<b>Name</b>	<b>Pacemaker Community Edition (CE)</b>
<b>Open Source</b>	Yes
<b>License</b>	proprietary
<b>Pricing</b>	n/a
<b>Magento Version</b>	>= 2.2.0
<b>Magento Edition</b>	Community + Commerce

## General

The **Pacemaker Community Edition (CE)** focuses on small and middle projects where import performance will not be a significant problem. As the **Pacemaker Community Edition (CE)** performance is already on a very competitive level, we're talking about medium projects when 100.000 **SKUs** and below have to be imported regularly.

Besides the **SKUs**, which are indeed a good benchmark, aspects like the number of websites, attributes, attribute options, or other technical requirements as the infrastructure that should be used can significantly impact performance.

Therefore the decision to use the **Pacemaker Professional Edition (PE)** instead of the **Pacemaker Community Edition (CE)** always needs to investigate deeper into a project's requirements and general conditions.

The **Pacemaker Professional Edition (PE)** is completely based on the **Pacemaker Community Edition (CE)** sources but replaces some main components to improve performance. Additionally, it comes with basic converter functionality and a **Redis** cache implementation that allows using a Redis server for caching.

It will be necessary for multi-process or -threaded environments to share state as well as global data.

## Standard Connector

### Operations

As the **Pacemaker Professional Edition (PE)** comes with a basic converter library that provides basic functionality to convert files from the source into the expected **CSV** format, an additional operation will be available.

As, in general, it will be possible to add any custom operation. This operation is virtual and needs to be implemented as needed

- convert

## Entity Types

### Products

As product import has the main focus in most cases, the **Import Library** focuses on improving performance, especially in that case.

- **Batch** import functionality (configurable)

- **Redis** cache adapter (configurable)

## Pacemaker Community Edition (CE) v3.7

<b>Hersteller</b>	TechDivision GmbH
<b>Web</b>	<a href="#">Pacemaker Community Edition (CE) v3.7</a>
<b>Open Source</b>	Yes
<b>License</b>	OSL 3.0
<b>Pricing</b>	Free Usage for <b>Magento Community</b> and <b>Commerce Edition</b>
<b>Magento Version</b>	>= 2.2.0
<b>Magento Edition</b>	Community + Commerce

- Common format and **MIME-Type** for comma separated values files **RFC 4180**
- Basic coding standard: **PSR-1**
- Logger interface: **PSR-3**
- Autoloading standard: **PSR-4**
- Caching interface: **PSR-6**
- Container interface: **PSR-11**

## General

The **Pacemaker Community Edition (CE)** focuses on small and middle projects where import performance will not be a significant problem. As the **Pacemaker Community Edition (CE)** performance is already on a very competitive level, we're talking about medium projects when 100.000 **SKUs** and below have to be imported regularly.

Besides the **SKUs**, which are indeed a good benchmark, aspects like the number of websites, attributes, attribute options, or other technical requirements as the infrastructure that should be used can significantly impact performance. Therefore the decision to use the Professional instead of the Community Edition always needs to investigate deeper into a project's requirements and general conditions.

The Professional Edition is completely based on the Community Edition sources but replaces some main components to improve performance. Additionally, it comes with basic converter functionality and a Redis cache implementation that allows using a Redis server for caching. It will be necessary for multi-process or -threaded environments to share state as well as global data.

## Technology

From a technical perspective, **Pacemaker Community Edition (CE)** can be used in two different ways. Either from within **Magento** or from outside.

By using it from within **Magento**, e.g. when the installation has been taken place as **Composer** dependency, simplifies usage in that way, that it will read most of the necessary configuration from the **Magento** installation itself.

With the use of outside middleware, those configurations has to be specified on the **CLI** or on parts of the configuration file.

- Installation via Composer or usage as PHAR
- PSR-3 compatible logger (configurable)

- PSR-6 compatible caching (configurable)
- Cache warming which is especially useful for slow DB connections (configurable)
- Archiving of dedicated import artifacts
- Fine-grained events to easily extend the standard functionality.
- Debug mode that ignores nonbreaking data inconsistencies
- Import to multiple Magento 2 installations/databases from one installation
- Zero-Effort extensibility for new columns that match the EAV attribute name for the entity types that are based on the Magento EAV model, which are
  - ☒ Categories
  - ☒ Products
  - ☒ Customers
  - ☒ Customer Addresses
- Easy mapping for columns that don't match the attribute code (by configuration)
- Build-In support for multi-process or -threaded environments
- Guarantee for consistent imports by
  - ☒ Supporting `.ok` files to mark import ready to be processed
  - ☒ creating PID files for each operation
  - ☒ moving files to a temporary directory
  - ☒ allowing to run operations within a Single Database Transaction
  - ☒ process multiple files (bunches) within one operation

## CLI (Command Line Interface)

**Pacemaker Community Edition (CE)** comes with a **CLI** application that allows to execute commands to achieve the expected result.

- Entirely based on **Symfony** following **PSR-1** and **PSR-4**
- Strictly using **Symfony DI** and supporting **PSR-11**
- Additional options to
  - ☒ Pass a custom configuration file
  - ☒ Pass a specific serial for the import process (mostly useful when using the **Import Library** in as a white-label solution)
  - ☒ Pass a specific **PID** file to avoid import collisions when running multiple import processes in parallel
  - ☒ Pass a custom **Magento** edition or version
  - ☒ Pass additional parameters or a file with additional parameters that will be used to customize standard functionality
  - ☒ Pass a custom **DB** connection
  - ☒ Pass custom source and target directory
  - ☒ Pass a custom archiving directory
  - ☒ Change the default log level
  - ☒ Activate the debug mode

- ☒ Activate the single transaction mode
- ☒ Deactivate the cache functionality
- ☒ Deactivate the archiving functionality
- Import commands for the available entity types
- Additional commands to
  - ☒ Create the `.ok` files
  - ☒ clear ghost **PID** files
  - ☒ export the configuration

## Workflow Engine (fully configurable)

- Uses **JMS** serializer for generic configuration parsing.
- Allows fine-grained configuration of every transaction.
- Each component consists of one or more separate libraries.
- Each component provides it's own configuration.

## Operations

An operation defines a set of functionality that will be executed during the import process.

Therefore the operation names reflect the import behavior that can be expected from it.

The following import operations are available.

- delete
- add-update
- replace

## Standard Connector

The standard connector is based on the **CSV** format as this is, in most cases, this is the least that can be delivered by third-party systems.

## Entity Types

The standard connector support Importing the following entity types.

## Products

The **CSV** file format of the product import is mostly compatible with the **Magento** standard **CSV** format.

## Product Types

The product import with the standard connector supports the following product types.

- Simple products
- Virtual products

- Configurable products
- Bundle products
- Grouped products

### **Additional Functionality**

Besides importing products itself, additionally, the following product-related data can be imported.

- Inventory < 2.3.1
- **MSI** > 2.3.1 (can also be part of the main product import)
- Tier prices (can also be part of the main product import)
- General Prices (also on Website Level)
- Media Gallery
- URL Rewrites
- Dynamic link types (upsell, cross-sell + related)
- Dynamic image types
- Dynamic creation of option values
- Translations on store view level

### **Clean-Up Functionality (add-update operation only)**

Nearly in every case, the add-update operation is the one that fits the requirements best.

One common restriction of this operation is that it only adds new or updates existing data.

As in many cases, it will be necessary to remove data.

The standard connector comes with a clean-up functionality for the most critical data.

The clean-up functionality can easily be activated/deactivated in the configuration.

- Product-category relations
- Product-website relations
- Empty columns (configurable)
- URL Rewrites
- Media gallery

## **Categories**

As **Magento** itself has no category import at all, the CSV file format is specific but uses the same approaches whenever possible.

### **General**

- Support for all category EAV attributes
- No entity ID necessary; instead, it uses the category paths

### **Additional Functionality**

- Translations on store view level
- Category Images
- URL rewrites

### Clean-Up Functionality (add-update operation only)

- URL rewrites

## Customers

**Magento** itself already provides customer import functionality. The **Import Library** functionality supports the same **CSV** format.

### General

- Support for all customer EAV attributes

## Customer Addresses

**Magento** itself already provides customers with address import functionality. The **Pacemaker Community Edition (CE)** functionality supports the same **CSV** format.

- Support for all customer **EAV** attributes

## Attribute Sets + Groups

**Pacemaker Community Edition (CE)** provides the possibility to import attribute sets and groups.

As **Magento** itself has no attribute set and group import, the format for the **CSV** file of the import is specific but uses the same approaches whenever possible.

### General

As the attribute import, the attribute set and group import are also not based on the **Magento EAV** model.

- Support for all attribute set and group attributes

### Additional Functionality

Create new attribute sets based on existing ones (configurable)

## Attributes

The **Import Library** provides the possibility to import attributes as well as their option values.

As **Magento** itself has no attribute import, the format for the **CSV** file of the attribute import is specific but uses the same approaches whenever possible.

### General

In contrast to the other entities, the attribute import is not based on the **EAV** model approach, as the different entity types are.

- Support for all attribute fields

- Support for all attribute types
- Sort order when rendering attribute sets and groups in the frontend/backend
- Option values for select and multi-select attributes
- Swatch values for text and visual swatch attributes

### **Additional Functionality**

Translation of labels and option/swatch values on store view level