Service Description - Pacemaker Enterprise Edition (EE) v1.3.0

Technische modul informationen

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

General

As processes are playing the leading role when it comes to **E-Commerce** in middle and enterprise scenarios, **Pacemaker** can be a solution for many of the existing and well-known problems.

We are talking here about issues like concurrency, in-transparency of errors, or data inconsistencies, which will be very expensive in the case they occur.

The primary purpose of **Pacemaker** is process modeling and coordination.

As **Pacemaker** has been built as a real **Magento** extension, it has the advantage in contrast to outside solutions that it has direct access to all running processes.

Therefore it will be possible to influence or, better to say, control them where and whenever necessary. It gives **Pacemaker** at least two options

- First, it can be a replacement for the impossible to control and monitor legacy **CRON** jobs
- Secondly, it can be something like a real **Pacemaker**, that controls the pace of the processes running within a **Magento** installation (which can and will be distributed among several hosts, in many cases)

Therefore **Pacemaker** will be distributed as set components, whereas each component has it is dedicated responsibility. The main component, which is indeed responsible for controlling the processes, is based on the **Pipeline** Pattern and strongly influenced by the **Gitlab Pipelines**.

Process Pipelines

Hersteller	TechDivision GmbH
Web	**Pacemaker Community Edition (CE)** v3.7
Open Source	Yes
License	Proprietary
Pricing	n/a
Magento Version	>= 2.2.0
Magento Edition	Community + Commerce

Compliance	n/a

General

Pipelines are a widely spread solution to allow stable, reproducible, and maintainable executing of processes.

Another advantage is the possibility of running pipelines in parallel, which will be necessary and very handsome in many others.

- Option to model nearly every process within a **Pacemaker** pipeline
- Possibility to breakdown monolithic legacy processes into small steps, which are following the single responsibility pattern
- Define conditions that decide whether or not a pipeline will be executed
- Define dependencies between processes and steps
- Monitor each process and its steps, either on the **CLI** or in the **Magento** admin **GUI**
- Download artifacts of a process or a step to a local/test instance in case of failures/problems to reproduce the behavior in a closed environment
- Optimize the utilization of the available resources to improve the system performance on the right places significantly

Ready-to-Use Pipelines

General

Besides the general functionality, one of **Pacemaker's** central added values comes ***out of the box*** is the process knowledge itself.

So it will be delivered with 3 ***ready to use*** pipelines that have been tested and are well known to work.

- Catalog import
- Price import
- Inventory import
- Order Workflow (Order Export / Status Update)

These ***ready to use*** Pipelines observe a configurable directory for a set of import files. Once the files are available, the import process will be initiated and executed.

These pipelines use **Pacemaker Community Edition (CE)** as an import library and provide the ability to import catalog data (attributes, attribute-sets, categories, and products), prices, and inventory (stock).

The predefined processes are highly extensible. There are many interfaces, which can be replaced by developers to change

- the pipeline steps
  - Change process order and behavior
  - Append or prepend additional tasks
  - Add further data transformation logic
- the file resolver
  - Change the name convention for import files

    - Define alternative file/directory structure
    - Use other file types
- the import library configuration
    - Add additional fields to import files
    - Adjust values (data mappings, etc.)

Pipeline Initializer

Pipeline Initializer is a process pipeline, which is responsible for the initialization of dynamic pipelines. Therefore it provides a simple PHP interface, which can be used in custom modules in order to start new pipelines with a dynamic set of steps.

Order Workflow

Order Export

The ready-to-use order export pipeline manages the export of orders.

It provides many extension points for customizations, allowing developers to offer their export and transport adapters.

Configuration

The order export can be configured for each website individually.

So different export formats and transport adapters can be chosen for each website and the filter criteria, which defines if an order is ready for export.

The default implementation allows multiple payments method and order status combinations as filter criteria.

Export Format

The export format is configurable for each website and can be easily extended by custom modules. By default, **Pacemaker** provides a file-based export, which can be configured as a template in **Magento's** admin **GUI**.

Transport Adapter

The transport adapter can also be configured for each website.

By default, **Pacemaker** provides a local file system adapter, which allows the export file to persist on any place within your web server.

Response Handler

Optional handler for external system response, asynchronous answers from **ERP** or **OMS**.

By default, there is a **JSON** file expected, which will be automatically executed.

This behavior can easily be changed with custom adapters.

Confirmation Mail Handler

Extension point to handle confirmation mail depending on **ERP/OMS** response. It is disabled by default.

Import Library v3.8

Hersteller	TechDivision GmbH
Web	**Pacemaker Community Edition (CE)** v3.7
Open Source	Yes
License	Proprietary
Pricing	n/a
Magento Version	>= 2.2.0
Magento Edition	Community + Commerce
Compliance	See **Pacemaker Community Edition (CE)** v3.8

General

The **Pacemaker Professional Edition (PE)** focuses on middle and large projects where import performance will become a significant problem.

As the **Pacemaker Community Edition (CE)** performance is already on a very competitive level, we're talking about middle projects when 100.000 **SKUs** and above 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.

Upload GUI

TBD

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 **Pacemaker Professional Edition (PE)** focuses on improving performance, especially in that case.

- Batch import functionality (configurable)
- Cache
- Cache Warming
- Change-Set Detection
- Timestamp Detection
- Extended Media (Video) Import
- Custom Product Option Import
- Redis cache adapter (configurable)

CLI (Command Line Interface)

The **Pacemaker Professional Edition (PE)** extends the **CLI** with additional commands and options

- Import commands for custom product options and dedicated media import
- Extend command for product import with options for change-set and timestamp detection

Import Library (CE) v3.8.0

Hersteller	TechDivision GmbH
Open Source	Yes
License	OSL 3.0
Pricing	Free Usage for **Magento Community** and **Commerce Edition**
Magento Version	>= 2.2.0
Magento Edition	Community + Commerce
Compliance	

- 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

In general, the **Import Library** has been implemented as a set of libraries on top of **Symfony** that can be used to implement custom import solutions for **Magento 2**.

It has been built to match the requirements of any presentable use case.

Besides the framework approach, it also provides a real implementation that improves integration, saves costs, and fits the requirements, at least, for many of the known use cases.

Technology

From a technical point of view, the **Import Library** can be used in two different ways.

Either from within **Magento** or from outside.

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.

Using it as a middleware, those configuration has to be specified on the **CLI** or must be part 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 **Magento EAV** model, which are
  - Categories
  - Products
  - Customers
  - Customer Addresses
- Easy mapping for columns that do not 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 executing the commands to achieve the expected result.

- Completely 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 **Pacemaker Community Edition (CE)** 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 operation
- 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 supports 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 will fit 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 address import functionality. The **Import Library** functionality supports the same **CSV** format.

- Support for all customer EAV attributes

Attribute Sets + Groups

Import Library provides the possibility to import attribute sets and groups.

As **Magento** itself has no attribute set and group import at all, 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

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

As **Magento** itself has no attribute import at all, 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