

The VILT logo is displayed in white capital letters within a white rectangular border. It is positioned over a blurred screenshot of a software interface with a blue overlay.

why  Motion ?

For more informations, send us an email to

motion.team@vilt-group.com

The VILT logo is shown in a light gray, semi-transparent style, consisting of the letters V, I, L, and T in a spaced-out font within a rectangular frame.

Motion is a framework
developed at **VILT**



Especially crafted to make migration processes easy. OpenText WEM Motion is an application built on top of a generic content movement engine designed to move and transform content across repositories.

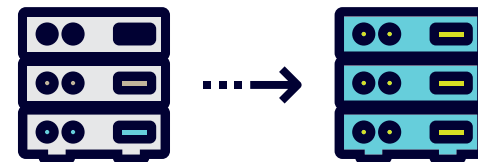
It has connectors for different types of repositories and is able to translate from one type of repository to another.

The screenshot shows the 'Migration' tab for job '#41 Export Site to Offline' on 10/02/2017 at 18:43:30. It indicates the job was created 4 minutes ago and is currently in progress at 17%. The progress bar shows 'Processing contents at level 0 (144 of 861)' with an execution rate of 9.9 Items/second. Below the progress bar are tabs for 'Stats', 'Items', and 'Plan'. A table lists the phases of the migration:

Phases	Start Date	Finish Date	Execution Time	Items Processed	Items Failed	Status
1. Compute	10/02/2017 - 18:43:30	10/02/2017 - 18:44:07	a few seconds	861	0	
2. Execution	10/02/2017 - 18:46:59			144	0	

so why **motion**?

- ∅ Migrate data between repositories
- ∅ Upgrade content repository versions
- ∅ Batch transformation in repositories
- ∅ Backup content data to disk
- ∅ Integrate with any repository
- ∅ Schedule data synchronizations
- ∅ Compare the content in two repositories
- ∅ Visual interface for selecting the content to export

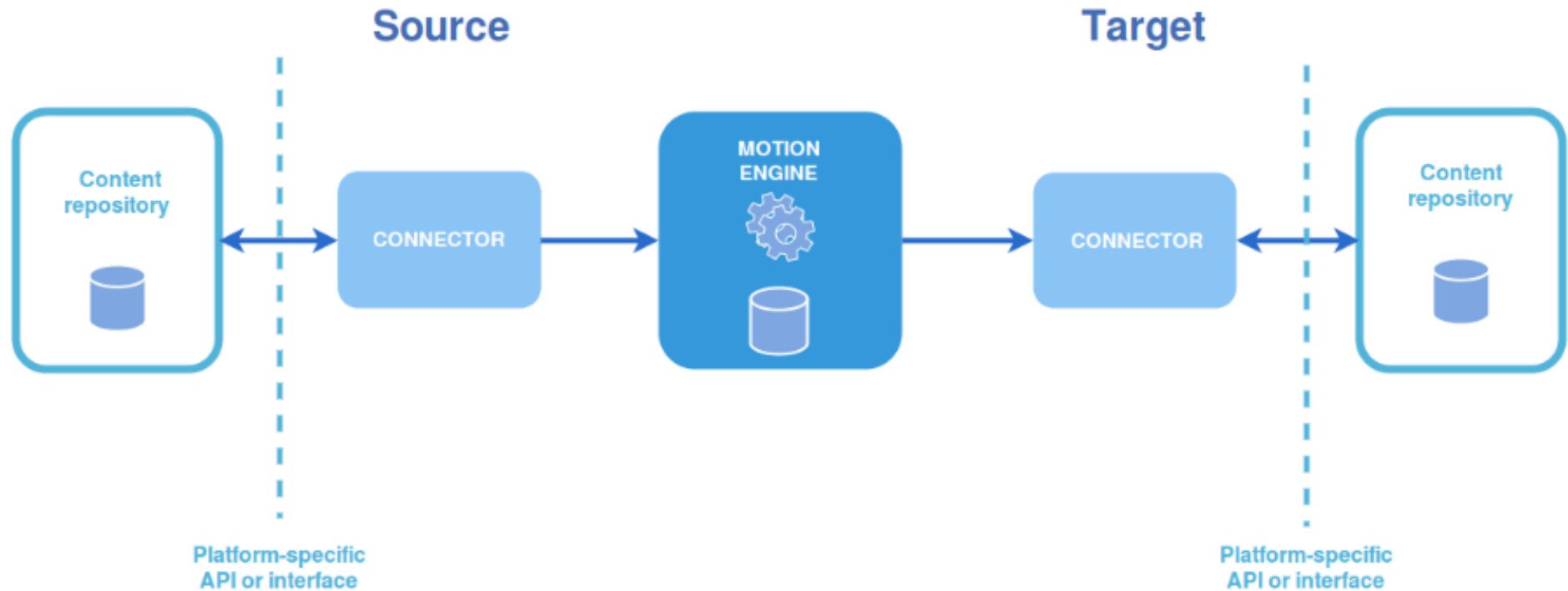


Motion is a framework developed at **VILT**



Minium is a framework
developed at **VILT**

let us explain **better**



- ∅ The Motion Connectors communicate with the Content repository
- ∅ The engine is agnostic of the source and target content repository
- ∅ Custom connectors can be implemented for your content repository

Application Servers

- ∅ Standalone (Spring Boot)
- ∅ Tomcat

Databases

- ∅ Oracle
- ∅ SQL Server
- ∅ PostgreSQL

how does it work?

Motion migration has 4 phases.

This allows you to validate everything you are exporting after the computation phase.

Performance-wise, it identifies items to be migrated in parallel, taking advantage of multi-thread migration.

During the transformation phase, custom transformation scripts can be applied to any content.

Finally all content can be imported to any connector that communicates with the Motion Engine

The entire process is streamlined, limiting the amount of memory and resources the engine needs to operate.

the 4 migration **phases**

∅ **Computation**

Graph is expanded given a set of root Items, and types of references to expand, and then Items are then organized in buckets

∅ **Export**

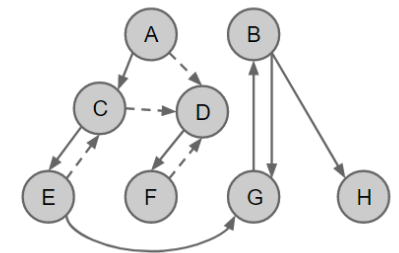
Items in the same bucket can be processed in parallel

∅ **Transformation**

Source connector provides the content data to import

∅ **Import**

Transformation scripts can be provided to change the data, target connector writes the transformed data



Motion is a framework
developed at **VILT**

? is that all? no! it also...

- ∅ Includes audit reports of your data migration
- ∅ Handles big content repositories with millions of items
- ∅ Solves complex referencing between contents
- ∅ Transforms your data with custom scripts
- ∅ Has a visual interface for comparing environments
- ∅ Schedules execution plans
- ∅ Has error analysis and recovery
- ∅ Can execute batch operations on your repository
- ∅ Compares the content in two repositories
- ∅ Allows content filtering with a visual interface or with scripting



new in **version 2**

- ∅ Ability to Cancel/Abort jobs
- ∅ Diagnostic and monitoring tools
- ∅ New unified interface
- ∅ Single process: No need to install a connector per stage.
- ∅ New architecture
- ∅ Faster Computation
- ∅ Faster import/read from source
- ∅ Smaller memory footprint
- ∅ Improved installation process
- ∅ Easy way to create motion transformation scripts in groovy with a syntax similar to jQuery
- ∅ Improved conflict resolution policies
- ∅ Audit and trace capabilities
- ∅ Comparison jobs
- ∅ Support for all WEM versions
- ∅ Support for PostgreSQL and TomEE



Motion is a framework
developed at **VILT**

Motion out-of-the-box comes with connectors for all WEM Versions.

This allows you to:

- ∅ Upgrade WEM installations to the latest release
- ∅ Move content across all stages
- ∅ Import/Export from management stage
- ∅ Export from production stage
- ∅ Synchronize Development and Production environments
- ∅ Run batch operations in Content Instances with Script Jobs
- ∅ Compare connector items between environments using the Comparison Jobs
- ∅ Move content from third-party repositories to WEM
- ∅ Backup content offline

motion for **wem**

The screenshot displays the Motion for WEM web interface. At the top, there's a navigation bar with links to Connectors, Navigators, Scripts, Jobs, and Executions. Below this, a filter section allows searching by Job Name. The main content area shows a list of migration jobs. Job #17, 'Migrate SiteA', is selected, showing its execution details. The job was created on 10/02/2017 at 12:10:54 and has an average execution rate of 25 items/second. It consists of two phases: '1. Compute' and '2. Execution'. The '2. Execution' phase is highlighted, showing it was completed with 2 failures. Below the detailed view, a list of other migration jobs is shown, including #7, #5, #4, and #3, all of which are completed.

Phases	Start Date	Finish Date	Execution Time	Items Processed	Items Failed	Status
1. Compute	10/02/2017 - 12:10:54	10/02/2017 - 12:10:55	⌚ a few seconds	25	0	
2. Execution	10/02/2017 - 12:10:55	10/02/2017 - 12:10:57	⌚ a few seconds	25	2	Completed with Failures

Supported Versions

- ∅ VCM 7.6 / 7.5
- ∅ WEM 16.2 / 16.0 / 10.5 / 8.5.1 / 8.5 / 8.1 / 8.0



Motion is a framework developed at **VILT**