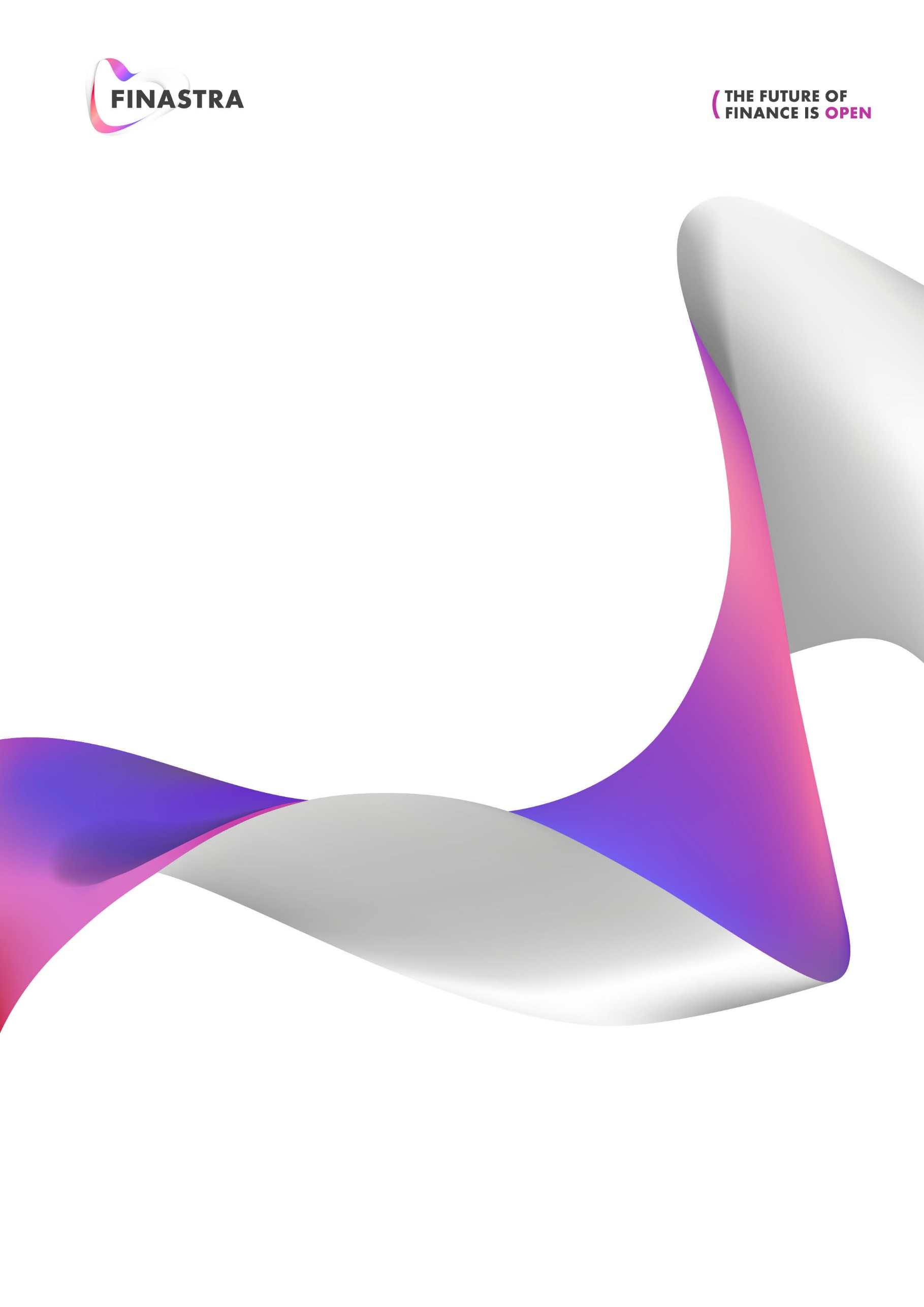
|  |
| --- |
| Su, Yuanjia  November 10, 2017 |



|  |
| --- |
|  |
|  |
| Mediolanum ETL User Guide |

CONTENTS

[Introduction 3](#_Toc498086267)

[Purpose 3](#_Toc498086268)

[Document Control 3](#_Toc498086269)

[ETL and ETL Chains 4](#_Toc498086270)

[User Guide 5](#_Toc498086271)

[Installation Guide 5](#_Toc498086272)

[Interfaces folder structure 5](#_Toc498086273)

[SophisETL.exe 5](#_Toc498086274)

[Configuration 5](#_Toc498086275)

[INI file Parameters 6](#_Toc498086276)

[XML file Parameters 6](#_Toc498086277)

[Mediolanum ETL Overview Schema 7](#_Toc498086278)

[1. Import RIMES benchmark components 7](#_Toc498086279)

[Objective 7](#_Toc498086280)

[Schema 7](#_Toc498086281)

[Extraction 8](#_Toc498086282)

[Transformation 8](#_Toc498086283)

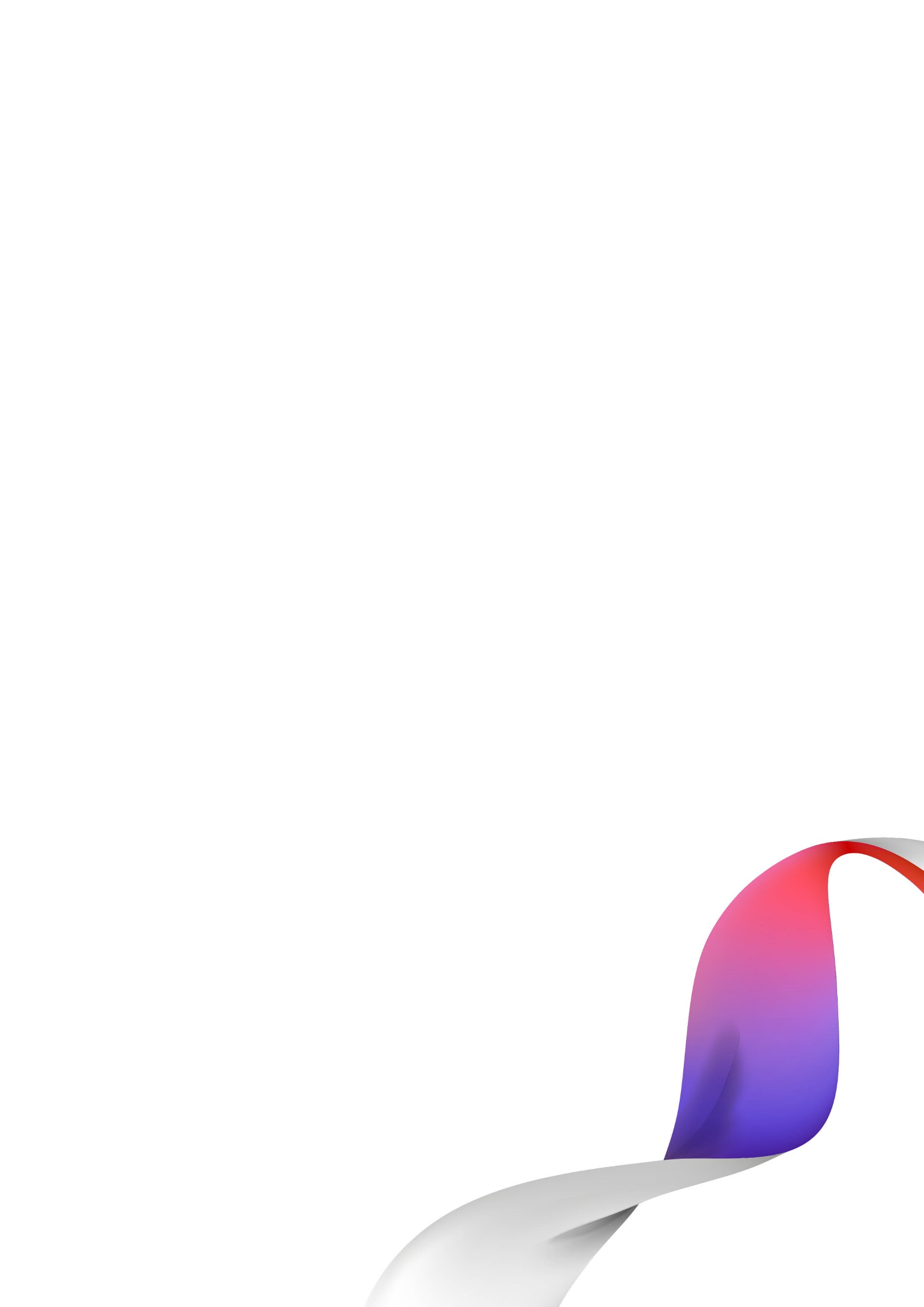
[Load 8](#_Toc498086284)

[Result 8](#_Toc498086285)

[2. Mediolanum PROD set up 8](#_Toc498086286)

[Input file 9](#_Toc498086287)

[Trouble shooting 9](#_Toc498086288)



# Introduction

## Purpose

This document aims to provide his reader with the keys to understanding and operating the ETL Chains developed during the FusionInvest implementation project. After a brief introduction on ETL concepts, we will address the setup of how to load RIMES benchmarks into FusionInvest. Each ETL procedure will be described as:

A conceptual diagram of its operations

A list of input and setup data

The results to expect

Some basic troubleshooting guidance

## Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Comments |
| 09/11/2017 | 1.0 | Finastra | Document Creation |

## ETL and ETL Chains

The term ETL covers a programming pattern designed to address the problem of data flows between heterogeneous systems. It keeps an abstract and modular approach to the necessary steps necessary to move data from a system to another.

Conceptually, moving data from System A to System B involves three kinds of operation:

Extracting the data from System A

Transforming the data from System A conventions into System B conventions, eventually enriching it in the process,

Loading the transformed data into System B

ETL is simply an acronym for this Extract – Transform – Load approach.

FusionInvest Sophis has developed its own implementation of the ETL pattern using the Steps (or Blocks) approach. A Step is responsible for a basic operation applied on a Data Record. There are three kinds of step:

Extract steps produce data records from a specific input stream (such as a CSV file, a relational Database, or a Web Service reply)

Transform steps acts directly on the flow of data records, changing them in the process (examples are a conversion to XML, grouping of records together, merging, filtering, mathematical operations on record fields, etc.)

Load steps consumes the data records and pushes them into a specific output stream

To keep a modular, brick-oriented approach, each step is as independent as possible from the data record content and performs only one task. Complex operations are performed by assembling multiple steps in a Chain. A Chain starts with one (or more) Extract steps, and finishes with one (or more) Load steps, with any number of Transform steps in the middle.

# User Guide

## 

## Installation Guide

ETL Interfaces can be run from any computer with Windows XP/Windows 7 or Windows Server installed

Installation Procedure:

1. Copy the installation folder into a local drive or network location. To run ETL Interface from a network location, the network location must be mapped to a network drive on the computer running ETL Interface.

2. Configure **parameter.ini** in **config** folder

## Interfaces folder structure

**import\_prices.xml**: This file is a template files used by **SophisETL.exe. SophisETL.exe** grabs text from template files, replace variables (text start with **$**) with their corresponding value, and send the text to Integration Service.

**parameters.ini**: specific configuration file for each interface (See Configuration section).

**Launch.SophisETL.bat**: script to execute the interface

**sophis\_etl.xml**: instructions to be performed by SophisETL.exe

## SophisETL.exe

**SophisETL.exe** located in **SophisETL** folder is a common binary executed by all interfaces. Upon finish execution, **SophisETL.exe** set the error level as following:

* **0** means no technical failure (does not mean 100% of records loaded)
* **1** means technical error (failed to connect to DB or IS, file not found…). Reports and logs should be investigated

Following table list parameters can be using with **SophisETL.exe**

|  |  |
| --- | --- |
| Parameter | Explanation |
| -X [xml\_file] | Define the xml file contains the instructions to be executed. By default, it is sophis\_etl.xml |
| -C [parameter\_file] | Define the configuration file |

## Configuration

All files with extension ini are configuration files.  
parameters.ini located in each interface folder used only by the interface. Parameters are replaced at run-time in the ETL definition when the keyword %%parameterName%% is used.  
Configuration file contains a number of lines. Each line is in the form on parameter=value. Lines starting with # are considered as comments.

### INI file Parameters

This chain depends on the following parameter being defined in INI file. You have to change all these values:

|  |  |  |
| --- | --- | --- |
| Parameter | Example | Explanation |
| csv\_file | Path to csv file. Support UNC, absolute and relative path | csv\_file containing prices |
| log\_file | Path to log file. Support UNC, absolute and relative path | File where logs will be filled. |
| report\_file | Path to report file. Support UNC, absolute and relative path | Contains success and/or issues for each imported prices. |

### XML file Parameters

This file contains some parameters which could be changed but it is optional. Default values are enough for test and production environments.

|  |  |
| --- | --- |
| Parameter | Explanation |
| configurationFile | Current value is “./SophisETL.exe.config” is the path of the Sophis configuration file containing Integration Service Login and Password filled by the SophisConfigurationManager.exe |
| reportType | Specify cases in which report will be filled. You can have the following values:   * allRecord : each records are reported * recordLoadedOnly: only successed are reported * recordNotLoadedOnly: only failure are reported |
| fields | Lines of the report can be customized. Thus you can specify a list of field that you want to display. In the current version I have chosen the 3 fields read from the csv file (ident, date, price).  Current version:  <fieldName>ID\_GP</fieldName> <fieldName>DT\_J0</fieldName> <fieldName>PX\_J0</fieldName> |

# Mediolanum ETL Overview Schema

## Import RIMES benchmark components

### Objective

### 

This chain is designed to update/insert historical benchmark components on a given date.

### Schema

EXTRACT

*CSV file reader*

TRANSFORM

*UpdateDateFormat*

LOAD

*ISLoad*

**Integration Service**

TRANSFORM

*ReplaceISINWithTicker*

TRANSFORM

*CSVToXML*

*Import Fixing*

**TEMPLATE**

*comp\_template.xml*

### Extraction

1. **ReadCSVData:** This chain reads only following columns in the csv file:

|  |  |  |  |
| --- | --- | --- | --- |
| CSV column | sophis\_etl.xml variable | | comment |
| VALUATION\_DATE | ValueDate | benchmark composition date | |
| INDEX\_CODE | BenchmarkRef | benchmark external reference ‘RIMES’ value | |
| MAIN\_CODE\_NAME | REFNAME | component reference type | |
| ISIN | ISIN | each underlying component | |
| BLOOMBERG\_TICKER | TICKER | ticker of each underlying component | |
| SECURITY\_WEIGHT | Weight | underlying composition weight | |

Sample csv:



### Transformation

1. **UpdateDateFormat**: Replace the value date format which is yyyymmdd by the one used by IntegrationService ie yyyy-mm-dd
2. **ReplaceISINWithTicker**: Unify the reference value to a single field
3. **CSVToXML**: The template XML named *comp\_template.xml* contains 5 variables which will be replaced by values contained in csv. These variables as the same name than csv columns.

### Load

**ISLoad**: the resulting XML record is booked in Sophis using the Integration Service. A Success or Failed status is added to the chain execution Report. You can also configure report only for issues.

### Result

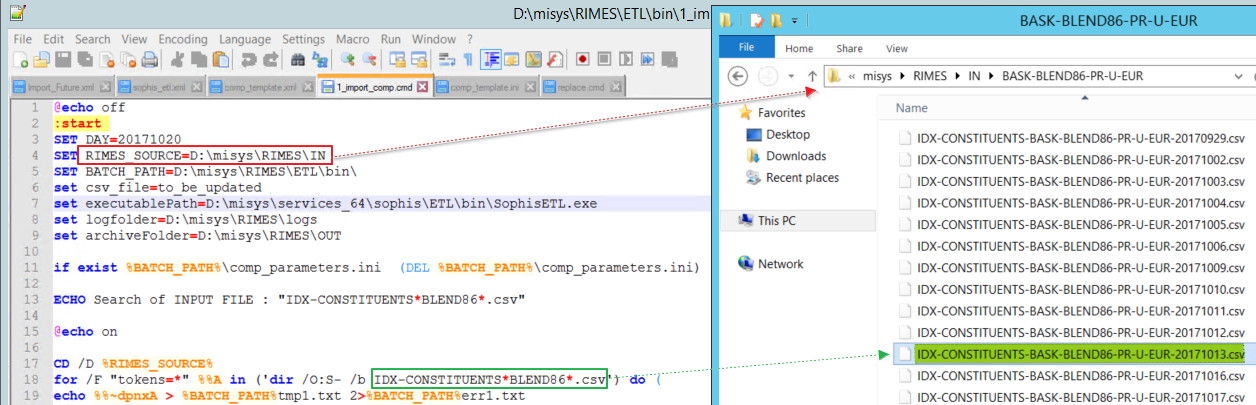
After processing, the benchmark contains composition of the value date.

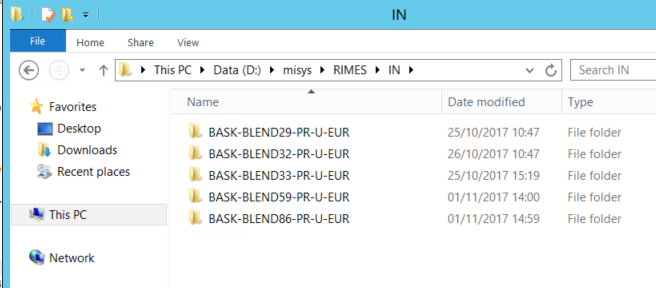
## Mediolanum PROD set up

The ETL process is triggered by a batch file, which can be found:

[\\10.9.244.102\d$\misys\RIMES\ETL\bin](file:///\\10.9.244.102\d$\misys\RIMES\ETL\bin) 

### Input file





The input folder variable is defined as **%RIMES\_SOURCE%** in the batch, where you will find a list of folders, one for each benchmark.

To correctly link an input file you want to import, simply alter the pattern as shown above **IDX-CONSTITUENTS\*BLEND86\*.csv** to match the csv file name. You can also check a temporal file generated on the fly under the main folder BATCH\_PATH%tmp1.txt to see if the file has been matched.

### Troubleshooting

Log files can be configured and found in the file comp\_template.ini under the main directory.

log\_file = D:\misys\RIMES\logs\comp\import\_comp.log

report\_file = D:\misys\RIMES\logs\comp\import\_comp.csv

It is advisable not to put the debug mode to true in PROD if you are processing enormous files daily. But if you do need to look into more details, make sure to set it to true in sophis\_etl.xml:

<debugMode>true</debugMode>

Most of the common issues (e.g. csv not found, chain setup incorrect, DB and/or IS connectivity issue) can be found in the log file. There are other issues that might come from Integration Service while loading, for instance faulty xml template, instrument not found etc. Such issues are normally logged in the report\_file.

E.g. [26/10/2017 15:39:03.787];ISLoad;2;Failure;GXML.0035 : NoSuchInstrument : No Instrument can be found with such a reference;No Such Instrument : FR0124423201;



|  |  |
| --- | --- |
| **About Finastra** Finastra unlocks the potential of people and businesses in finance, creating a platform for open innovation. Formed in 2017 by the combination of Misys and D+H, we provide the broadest portfolio of financial services software in the world today –spanning retail banking, transaction banking, lending, and treasury and capital markets. Our solutions enable customers to deploy mission critical technology on premises or in the cloud. Our scale and geographical reach means that we can serve customers effectively, regardless of their size or geographic location – from global financial institutions, to community banks and credit unions. Through our open, secure and reliable solutions, customers are empowered to accelerate growth, optimize cost, mitigate risk and continually evolve to meet the changing needs of their customers. 48 of the world’s top 50 banks use Finastra technology. Please visit [**finastra.com**](http://www.finastra.com)  Finastra and the Finastra ‘ribbon’ mark are trademarks of the Finastra group companies.  © 2017 Finastra. All rights reserved. | **Corporate Headquarters** One Kingdom Street Paddington London W2 6BL United Kingdom  T +44 20 3320 5000 |
|  |