 User Guide



ETL Chains

This document describes the ETL Chains developed for Tikehau for Price data Integration

Doc_Cover_Page_Banner_Flipped

January 2013

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Modification |
| 16/01/2013 | 1.0 | Sophis (Fabrice) | Document Creation |

Table of Contents

[Introduction 3](#_Toc287351784)

[Objectives 3](#_Toc287351785)

[ETL and ETL Chains 3](#_Toc287351786)

[User Guide 4](#_Toc287351787)

[Presentation 4](#_Toc287351788)

[Installation Guide 5](#_Toc287351789)

[Configuration 5](#_Toc287351790)

[Interfaces folder structure 6](#_Toc287351791)

[Logs and Reports 7](#_Toc287351792)

[SophisETL.exe 8](#_Toc287351793)

[Interface Execution Detail 9](#_Toc287351794)

[Chain I6: Cash True-Up 9](#_Toc287351795)

[Chain I7A: Unit Prices 14](#_Toc287351796)

[Chain I7B: Unit Quantities 18](#_Toc287351797)

[Chain I8: Level 1 Units FX Exposure 23](#_Toc287351798)

[Chain I9: MBL Margin 27](#_Toc287351799)

[Chain I10: Level 3 Daily Cash Flows 32](#_Toc287351800)

[Chain IX1: FX Collateral 37](#_Toc287351801)

[Chain IX2: Goods and Services Tax Receivable True-Up 42](#_Toc287351802)

[Chain IX3: Benchmark Sector Composition 47](#_Toc287351803)

[Chain IX4: Historical Price Uploader 51](#_Toc287351804)

# Introduction

## Objectives

This document aims to provide his reader with the keys to understanding and operating the ETL Chains developed during the Sophis Value implementation project.

After a brief introduction on ETL concepts, each chain 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

## 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.

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> |
|  |  |

## ETL Overview Schema

### Objective

This Chain read only some columns in the csv file:

* + ID\_GP: Identifier of the instrument
  + DT\_J0: The price date
  + PX\_J0: The price value

### Reports

All interfaces write report of each run in Reports folder. Reports should be checked after each run to make sure the execution successful. If any error was reported, Logs folder can be referenced to find the root cause.

### Schema

EXTRACT

*CSV file reader*

TRANSFORM

*DateFormater*

LOAD

*ISLoad*

**Integration Service**

TRANSFORM

*ReplacePrice*

TRANSFORM

*CSVToXML*

*Import Fixing*

**TEMPLATE**

*import\_prices.xml*

### Extraction

1. **ReadCSVData:** This Chain read only some columns in the csv file:
   1. ID\_GP: Identifier of the instrument
   2. DT\_J0: The price date
   3. PX\_J0: The price value

### Transformation

1. **DateFormater**: Replace the price date format which is dd/mm/yyyy by the one used by IntegrationService ie yyyy-mm-dd
2. **ReplacePrice**: Replace csv file price number using “, “ as separator by “.” which is used Integration Service.
3. **CSVToXML**: The template XML named *import\_prices.xml* contains 3 variables which will be replaced by values contained in csv. These variables as the same name than csv columns.

### Load

**IntegrationServiceImporter**: 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 instrument contains the new price.