Nasdaq Calypso

Markit Valuation Service Interface

Version 5.0.3

Revision 1.0
February 2022
Approved

Document History

Revision	Published	Summary of Changes
1.0	July 2011	New Document
2.0	December 2011	Added scheduled task name, added the details for setting valuation date offset for the Import valuation Scheduled task
3.0	January 2012	Added error identification in case of failures in uploading trades for valuation to MarkitPV
4.0	May 2012	Added Release notes for May 2012 Service Pack
5.0	July 2012	Added section 2.2.4.1 Proxy support, 2.2.4.2 Output file name and 4.2 Changes in July service pack
6.0	August 2012	Added EquityStructuredOption and Equity Forward to the supported product list. Added release notes for Aug Service pack
7.0	September 2012	Added release notes for Sept service pack
8.0	November 2012	Added release notes for Oct service pack, Nov service pack
9.0	May 2013	Added release notes for May service pack
10.0	July 2013	Added release notes for July service pack
11.0	Nov 2013	Added release notes for Nov service pack 2.4.1
12.0	Jan 2014	Added release notes for Jan SP 2.4.2
13.0	Feb 2014	Added release notes for Feb SP 2.4.3
14.0	Mar 2014	Added release notes for Mar SP 2.4.4, Mar SP 2.4.5
15.0	Apr 2014	Added release notes for Apr SP 2.4.6

Revision	Published	Summary of Changes
16.0	Jun 2014	Added release notes for Jun 2014 SP 2.5
17.0	Aug 2014	Added release notes for Aug 2014 2.5.1
18.0	Sep 2014	Added release notes for Sep 2014 2.5.2
19.0	Dec 2014	Added release notes for Dec 2014 2.5.3
20.0	Jan 2015	Added release notes for Jan 2015 2.6.0
21.0	Feb 2015	Added release notes for Feb 2015 2.6.1, Feb 2015 2.6.2
22.0	Mar 2015	Added release notes for Mar 2015 2.7.1, 2.9.1
23.0	June 2015	Added release notes for Jun 2015 2.10.0
24.0	Mar 2016	Added release notes for Mar 2016 2.15.2
25.0	September 2018	Added release notes for Sep 2018 3.4.2
26.0	October 2018	Added release notes for Oct 2018 2.11.1
27.0	November 2018	Updates for version 3.7.0
28.0	January 2019	Updates for version 3.4.3
29.0	February 2019	Updates for version 3.7.2
30.0	March 2019	Updates for version 3.4.4, 3.8.0
31.0	May 2019	Updates for version 3.4.5
32.0	June 2019	Updates for version 3.8.1
33.0	January 2020	Updates for version 3.4.6
34.0	February 2020	Updates for version 4.0.1
35.0	May 2020	Updates for version 4.0.2
36.0	July 2020	Updates for version 4.0.3
37.0	August 2020	Updates for version 4.0.4
38.0	September 2020	Updates for version 3.4.7, 3.4.8, 3.4.9, 4.1.0, 3.8.2
39.0	October 2020	Updates for version 4.2.0
40.0	November 2020	Updates for version 4.3.0

Revision	Published	Summary of Changes
41.0	December 2020	Updates for version 3.4.10, 4.3.1
42.0	February 2021	Updates for version 4.4.0, 4.4.1
43.0	March 2021	Updates for version 3.4.11
44.0	April 2021	Updates for version 4.4.2
45.0	May 2021	Updates for version 3.4.12, 4.5.0
46.0	June 2021	Updates for version 3.4.13, 4.5.1
47.0	July 2021	Updates for version 4.4.3, 4.6.0.
48.0	August 2021	Updates for version 4.7.0, 4.7.1.
49.0	September 2021	Updates for version 4.8.0.
50.0	October 2021	Updates for version 4.9.0. Release Notes are now provided in the PBFA Module Release Notes, and Core Calypso Monthly Release Notes.
51.0	February 2022	Updated for version 5.0.0, 5.0.1, 5.0.2, 5.0.3 - - Technical release only – Version 17.0 compatibility

This document describes the interface between Calypso and the Markit Valuation Service.

Table of Contents

Introduction

Markit provides financial institutions with an independent valuation service for their portfolio of trades. This service is a Web-based platform enabling clients to upload their trade data and then retrieve corresponding valuation measures for each trade.

The need has come from several Calypso clients to automate the use of this service by having functionality in Calypso to upload the trade data to Markit and download the resulting valuations.

The following sections in this document will describe Calypso functionality to address this need.

Installation

This section applies to all PBFA interfaces:

- Markit Valuation Service
- HSBC
- Morgan Stanley
- Northern Trust
- SEI

The PBFA interfaces are installed as part of the Calypso Installer when you select the Markit Present Value interface and the Data Uploader optional module.

Interface Design

3.1 Scope

Products supported by the interface include the following:

- CDS
- CDX
- CDSLoan
- CDX Tranche
- Bond
- Bond Asset Backed
- BondBrady
- IR Swaption
- Caps/Floors
- FX
- FX Forward
- FX Swap
- FXNDF Swap
- Swap
- Cross Currency Swap
- Equity Swap
- Total Return Swap/ Performance Swap
- Credit Default Swaption
- CDSIndex Option
- FRA
- Inflation Swap
- FX Option
- Equity Structured Option
- Equity Forward
- CMS Swap
- Non Deliverable Swap
- Dividend Swap

3.2 Functional Analysis

The new exchange functionality consists of two scheduled tasks:

- One for exporting trade data to Markit website (INTERFACE_FEED).
- One to import Markit valuations for these trades into Calypso (INTERFACE_CONFIRMATION).

3.2.1 Assumptions

- User has the valid Markit User Name and Password to upload Trades and Download File (This needs to be Configured on The Scheduled Task)
- All Uploading and Download to Markit is done in the XML format.
- All the Upload / Download with Markit is over HTTP protocol (Markit Pricing supports HTTP Protocol)
- User has datauploader - jar in classpath. These are part of the package delivered. Note that if Uploader is used for some other purpose(other than MarkIT PV), it needs to be licensed separately

List of jars to be in classpath:

- cal-upload.jar
- datauploader-xxx.jar
- edtftpj-pro-xxx.jar
- sjsxp-xxx.jar
- pbfa-xxx.jar

3.2.2 Configuration

Add the following in the service.properties (for the New RMI Service that contains the API to retrieve Mapping Values).

```
services=AccessServer,TradeServer,BackOfficeServer,ReferenceDataServer,Acc
ountingServer,ProductServer,MarketDataServer,FXDataServer,CommodityServer, DataUploadServer



DataUploadServer.name=DataUploadServer DataUploadServer.class=com.calypso.tk.service.DataUploadServerImpl
DataUploadServer.port=<port_no>

DataUploadServer.server- handlers=com.calypso.tk.service.handler.TransactionInvocationHandler,com.c
alypso.tk.service.handler.ServerInvocationHandler

DataUploadServer.client- handlers=com.calypso.tk.service.handler.ClientInvocationHandler
```

If you are using Calypso Version 12 or above the same goes in the resources/appconfig/Core.DataServer.serviceconfig.xml

```
<bean id="baseDataUploadServer" class="com.calypso.tk.service.DataUploadServerImpl">

</bean>

<bean id="rmiBaseDataUploadServer" parent="rmiServiceExporter">
```

```
<property name="service" ref="baseDataUploadServer" />
<meta key="serviceInterface" value="com.calypso.tk.service.RemoteDataUpload" />
</bean>
```

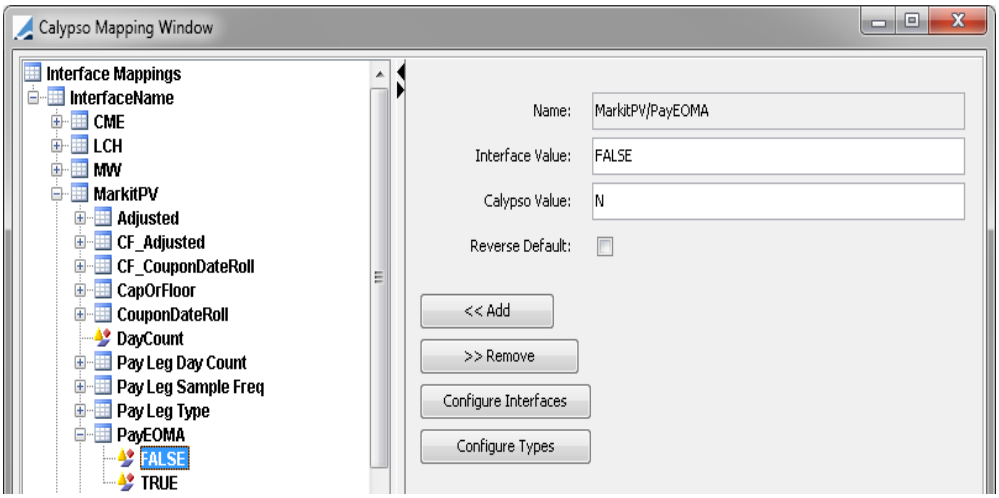
All the domain values and mapping types are inserted in the database by running the Execute SQL. In the Execute SQL window, the data files should already be loaded: “SchemaBase.xml” (found in release), “GatewaySchemaBase.xml” (found in datauploader) and “PBFA_MarkiTPV_SchemaData.xml”.

Mapping From Calypso values to the MarkIT PV values:

All the columns / types of calypso that have different values at MarkITPV side need to be mapped with corresponding markit values from the Calypso Mapping window.

Navigate to the Calypso Mapping window using **Processing > Tools > Calypso Mapping** (menu action `mapping.CalypsoMappingWindow`) from the Calypso Navigator.

The following screenshot shows the calypso mapping window:



Rate Index Mapping

If trade keyword PlatformContractualDefinition = ISDA2021 is set on the Trade, the RateIndex_ISDA2021 mapping is used to determine the rate index.

Example:

Name = MarkitPV/RateIndex_ISDA2021

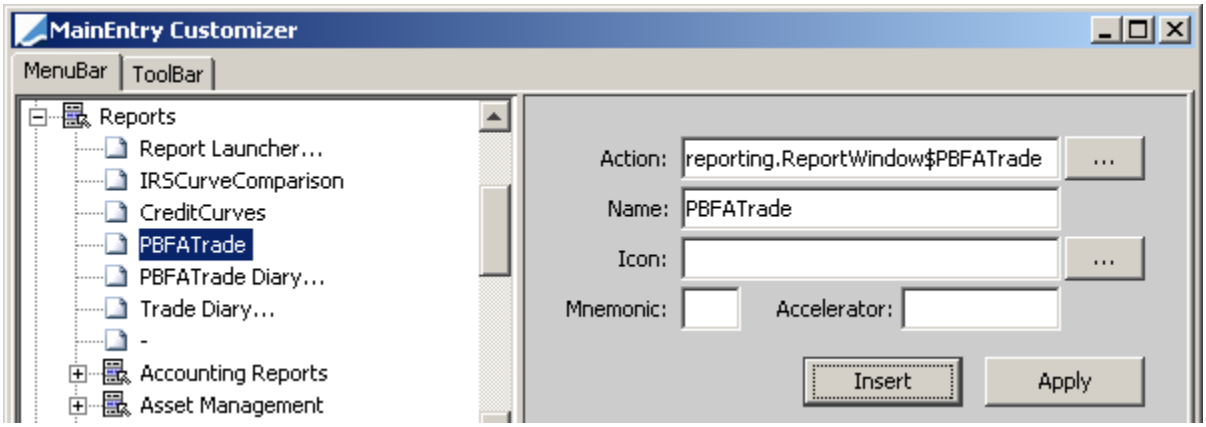
Interface Value = USD-SOFR-OIS Compound

Calypso Value = USD~SOFR~COMPOUND

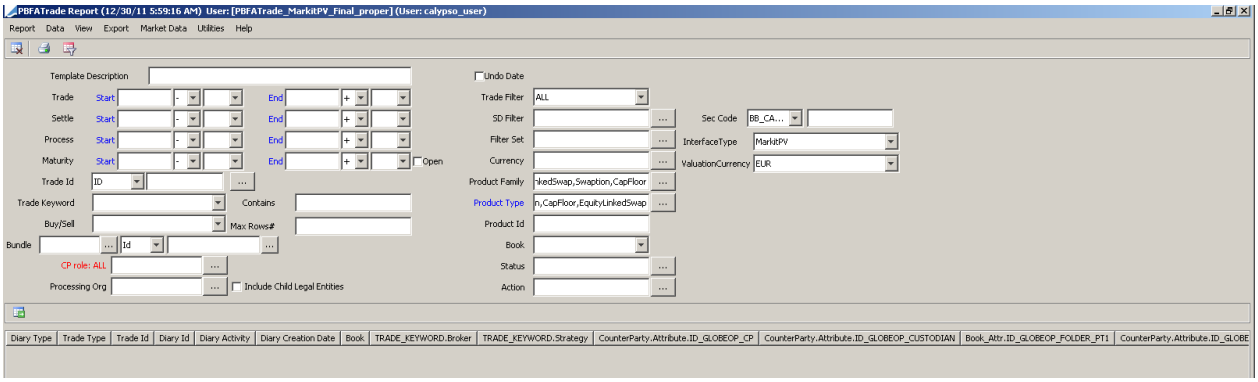
This applies to the following products: Vanilla IRS, Xccy Swap, CMSSwap, Swaption, InflationSwap.

3.2.3 Import Report Template

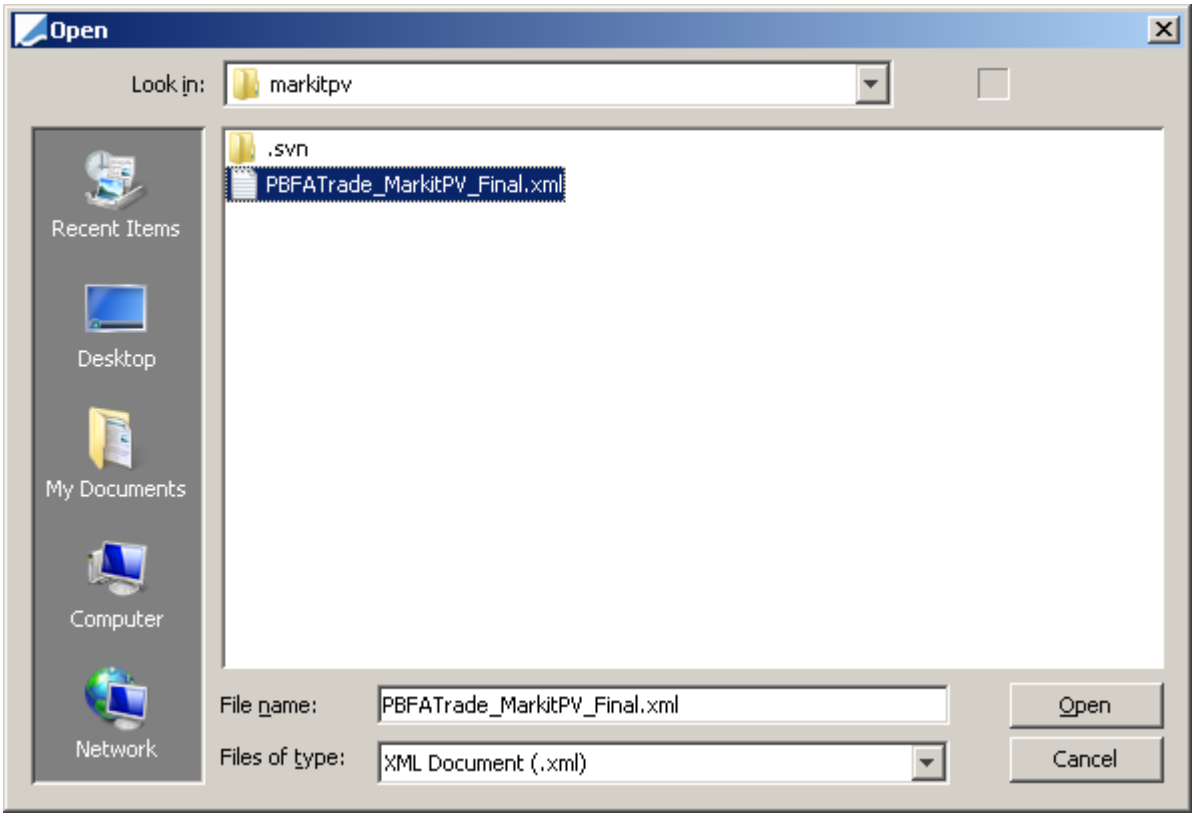
Configure Report Window from the Main Entry Configurator as follows.



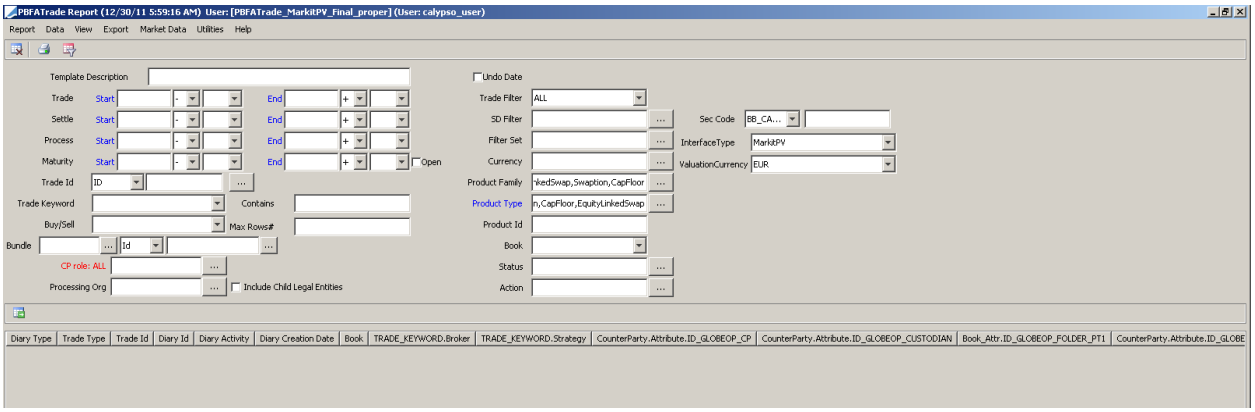
Launch the Report window.



Import the Template (resources/calypso/mapping/markitpv/PBFATrade_MarkitPV_Final.xml) from the menu **Utilities > Import Template**.



Template imported.



Please make sure that the interface type is MarkiPV and select the desired valuation currency from the dropdown.

3.2.4 Exporting - Scheduled Task INTERFACE_FEED

This scheduled task uploads the Trade Report Data to Markit.

It is used with a trade filter to select the trades to be sent to Markit and be valued.

When you run this task, you can choose the trades from a trade filter or by passing dates as described below. If using scheduled task, create the trade filter first and then use it from the Scheduled Task Window and Save that task.

The scheduled task uses the following attributes to describe where and how the xml file will be sent.

Attribute	Description
Interface Name	Always “MarkitPV”
Instance	Give an instance name if multiple instances of same scheduled task need to be run.
Output Folder	C:/Test (To generate Temp Files to Send to markit)
Output File Name	Set the prefix for the output file name.
Report Type	PBFATrade
Template Name	Name of the Template
Start Date	Start Date Empty when Using Trade Filter (or) for trades created today.
End Date	End Date Empty when Using Trade Filter (or) for trades created today.
TransportType	HTTP (Markit PV supports https and setting this field to HTTP supports both http and https)
HostName	pv.markit.com
UserName	Login Name
Password	Login Password
Destination	https://pv.markit.com/upload
Load Type	<Leave Blank>
Proxy Type	DIRECT or PROXY (Please set this flag to proxy in case of proxy connection)

Proxy Server Parameters

The calypso MarkitPV interface supports a proxy based connection. It is configurable on the scheduled task window via the “Proxy Type” attribute as shown in the below screenshot:

As shown in the screenshot it has two modes: DIRECT/PROXY:

- DIRECT - Represents a direct connection, or absence of proxy.
- PROXY - Represents a proxy using the HTTP protocol.

Besides above parameter, you need to define the following proxy parameters in the Calypso Environment properties in order to do a connection via proxy:

- HTTP_PROXY_HOST
- HTTP_PROXY_PORT
- HTTP_PROXY_USER (Optional)
- HTTP_PROXY_PASSWORD (Optional)
- HTTP_PROXY_DOMAIN (Optional)

The requests will be routed via the proxy server to MarkitPV if the above parameters are configured.

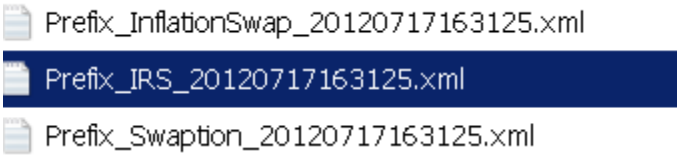
Output File Name

The output file format is described in the Markit Portfolio Valuations XML guide attached in the Appendix at the end of this document.

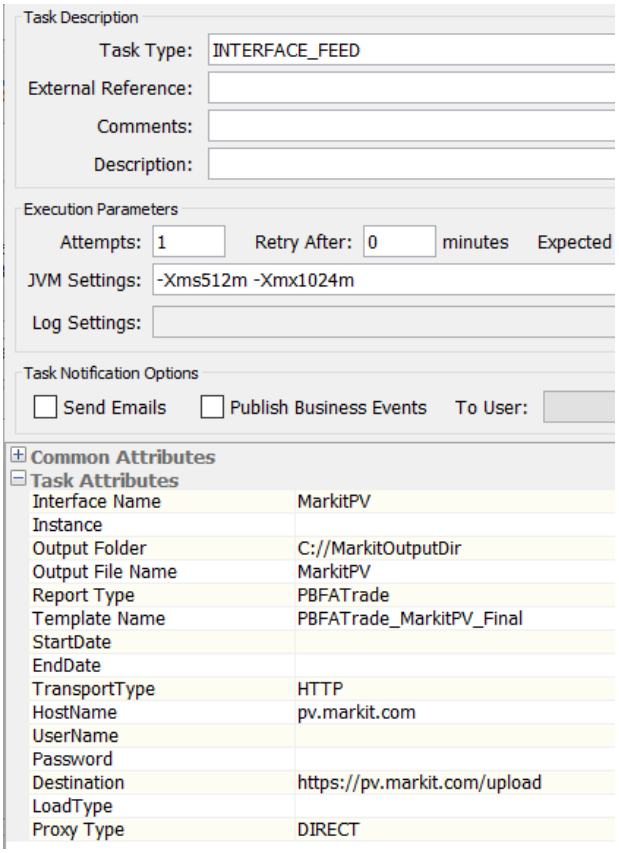
The transformed files will be saved in the output directory specified in the scheduled task attribute “Output Folder” having the file name prefix specified in the scheduled task attribute “Output File Name”.

Output Folder	c://MarkitOutputDir
Output File Name	Prefix

The resultant file name will have the “<prefix_specified>_<productName>_<timestamp>.xml”. The below screenshot shows a sample file name.



Below screenshot shows the scheduled task window showing the INTERFACE_FEED scheduled task.

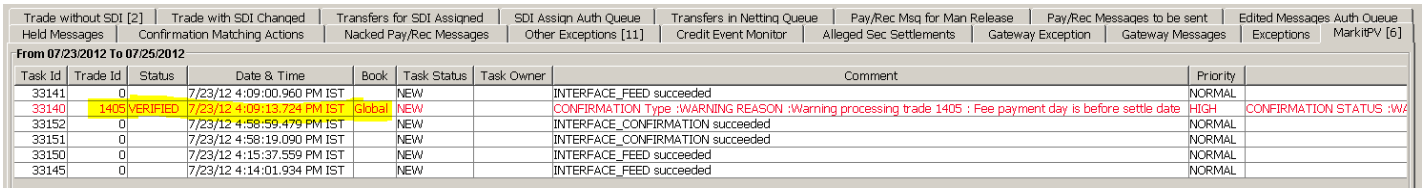


If you want to check if there are any errors during the trade upload to MarkitPV site using the INTERFACE_FEED scheduled task, you can check from the Task Station. The Task Station displays all the validation errors that occurred while sending the trades to MarkitPV for the following types of exceptions:

EX_INFORMATION

EX_INTERFACE_EXCEPTION

Example:



3.2.5 Importing Valuations and Markit Sensitivities - Scheduled Task INTERFACE_CONFIRMATION

This scheduled allows importing trade valuation data and risk sensitivities from Markit to Calypso. All the Valuations use PricerFromDB to “Save” valuations in Calypso.

The scheduled task uses the following attributes to describe where to get the files and for which valuation date. It also defines where to save the files locally in addition to store them in the Calypso database.

[Note: If you wish to import valuations from the locally available file on file system, add value 'LOCAL' to domain 'InterfaceTransportTypes'.]

Attribute	Description
Interface Name	MarkitPV
Type	MarkitValuations, MarkitSensitivities (risk sensitivities)
TransportType	HTTP (or LOCAL)
HostName	pv.markit.com
UserName	Markit Login Name
Password	Markit Login Password
Destination	https://pv.markit.com/download
Working Folder	<User defined>
LocalValuationsFile	When transportType = 'LOCAL', specify local valuation file location and file name Not used for MarkitSensitivities
TradeAction	Not used
TradeFieldName	Not used
KeywordName	Not used
Proxy Type	DIRECT or PROXY (Please set this flag to proxy in case of proxy connection)

The scheduled task downloads the valuations as of the current date by default. If you need to download the valuations as of previous date, you need to set the appropriate value in the "Valuation Offset" field in the scheduled task window. You also need to specify the holiday calendar to be considered by the scheduled task.

3.2.6 Storing Valuations in Calypso Database

The Trade_Price table in the Calypso database is dedicated to storing external valuations and will be used to store and retrieve Markit valuations. Here is a description of this table:

TRADE_PRICE	Stores trades prices calculated by Calypso or by external systems using the scheduled task EOD_TRADE_VAL_DB. Such prices can be used to price trades using the pricer PricerFromDB. *Java Class – com.calypso.tk.util.sql.TradePriceSQL*			
TRADE_PRICE_HIST	Archive table for trade_price.			
Column	**Type**	**Size**	**IsNull**	**Description**
trade_id	numeric	18	N	Unique number to identify trade.
valuation_date	datetime	23	N	Valuation date.
measure_id	numeric	18	N	Unique number to identify pricer measure.
measure_value	double precision	15	Y	Pricer measure value.
currency_code	varchar	3	Y	Unique code to identify currency such as EUR, USD.

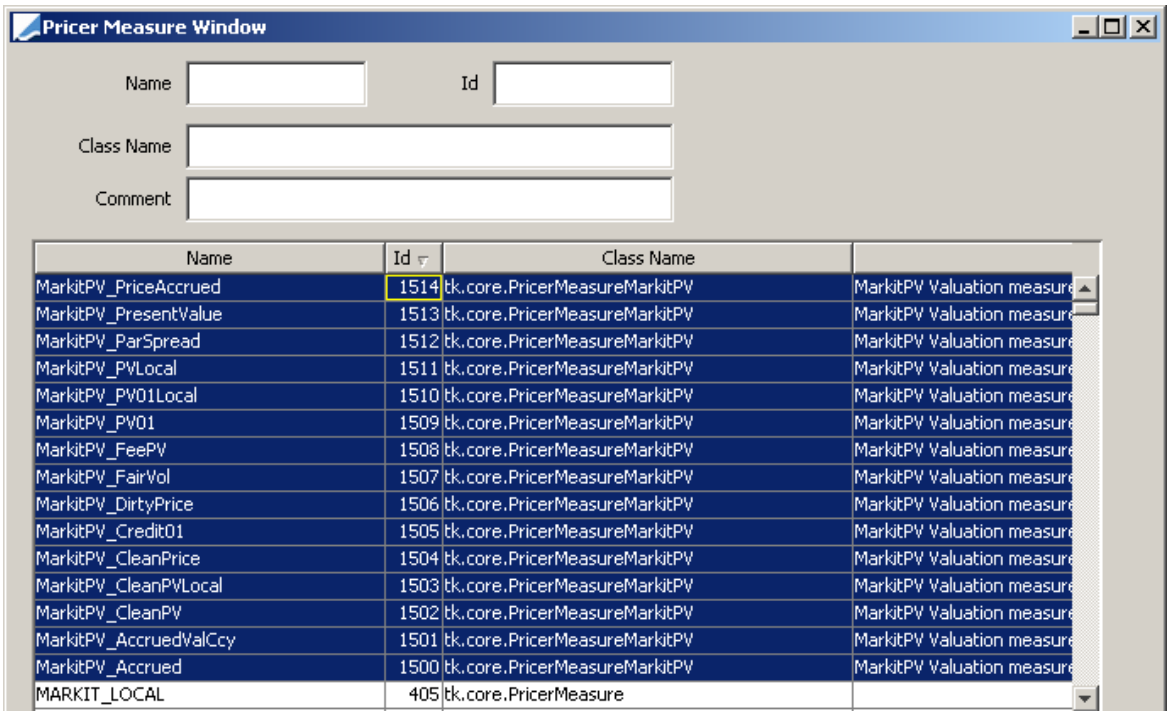
The Pricer Measures can be viewed from the Calypso Navigator under Configurations > System > Add Pricer Measure.

It shows the following window.

Custom pricer measures are created in Calypso to store valuations coming from Markit. They are prefixed with MarkitPV_ to differentiate them from out-of-the-box pricer measures.

[NOTE: These measures are visible in all places where Calypso pricer measures are visible]

In the example below, the custom measure MarkitPV_PVLocal contains the PVLocal valuation coming from Markit.



The measure naming convention needs to be followed. The convention is <InterfaceName>_measureName. So for MarkitPV interface it is “MarkitPV_NPV” for example. While adding the measure, the following needs to be added:

- MEASURE_NAME - MarkitPV_NPV
- MEASURE_ID - Number (one greater than the last id present)
- CLASS_NAME - tk.core.PricerMeasureMarkitPV
- COMMENT - Description of the measure

The interface automatically creates the Pricer Measures (shown above in the screen).

All the Pricer Measures coming from Markit are saved into a domain value `MarkitPVPricerMeasures` (these are automatically populated when you run Execute SQL).

When you run the INTERFACE_CONFIRMATION scheduled task, the interface connects to markit, downloads the valuations file, and processes it as follows:

- Checks if the Pricer Measure found in the valuations file is available in Calypso.
- If not then checks to see if the Pricer Measure is listed in the domain value specified above, and then create the Pricer Measure automatically in calypso.

However, while creating the Pricer Measure, the ID must be unique for each Pricer Measure, so another domain `MarkitPVPricerMeasureIndex` (automatically populated when you run Execute SQL) contains the base value: 1500.

So, when the Interface creates the pricer measure it reads this domain and then uses this ID, then checks if this ID is used for any other Pricer Measure in the system. If so, it Increments and checks again, this continues until a unique id is found.

The name of the corresponding tag in the xml is derived from the measure name. So for measure name “MarkitPV_PVLocal”, the framework searches for corresponding <NPVLocal> tag in the valuations xml and save the valuation in the database against the measure name “Markit_NPVLocal”.

This custom pricing measure will be used to store the PVLocal valuation present in the Markit xml:

```
<value>
<TradeId>1234</TradeId>
<Account>MarkIt</Account>
<Book>Test</Book>
<LegId>2</LegId>
<Notional>1000000</Notional>
<LocalCcy>GBP</LocalCcy>
<ValuationCcy>USD</ValuationCcy>
<Status>Succesful</Status>
<PVLocal>-3415100.3996771094</PVLocal>
```

```
<PresentValue>-6796903.570457367</PresentValue>
<Accrued>-327250.26739726026</Accrued>
<PV01>0</PV01>
<CleanPVLocal>-3177850.132279849</CleanPVLocal>
<CleanPV>-6324716.22576997</CleanPV>
```

Once the INTERFACE_CONFIRMATION scheduled task is completed all the valuations from Markit are stored in Calypso database in the TRADE_PRICE table. You can use the Trade Price Report to view the pricer measures.

3.2.7 Custom Pricing Measures Examples

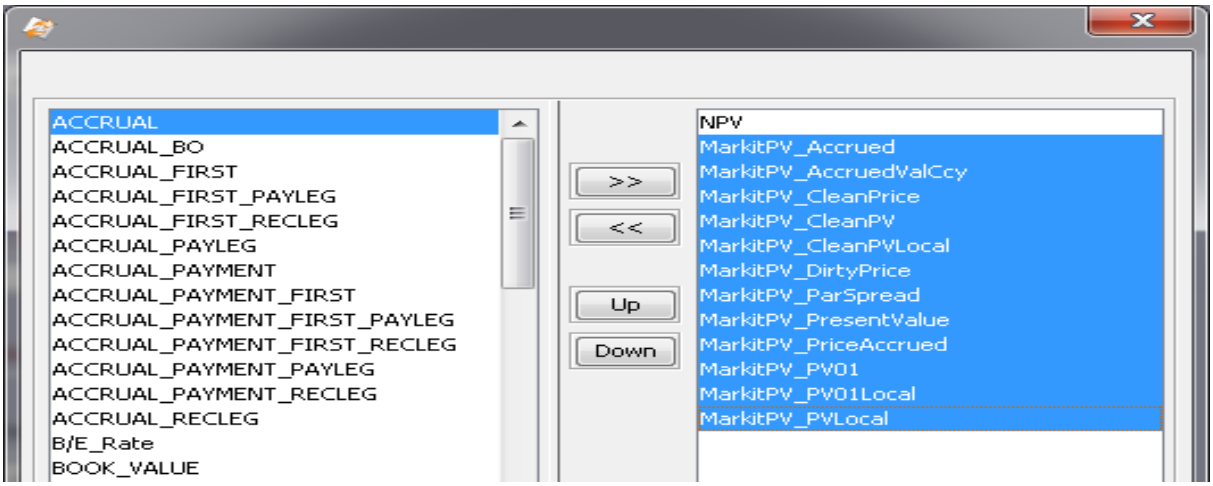
Markit has its own Pricer Measures, and these are Different from the Measures available in Calypso.

Following are the examples of Custom Measures:

Markit Measure	Custom Markit Measure in Calypso Measure
PVLocal	MarkitPV_PVLocal
PresentValue	MarkitPV_PresentValue
PV01	MarkitPV_PV01
Accrued	MarkitPV_Accrued
CleanPVLocal	MarkitPV_CleanPVLocal
CleanPV	MarkitPV_CleanPV
AccruedValCcy	MarkitPV_AccruedValCcy
CleanPrice	MarkitPV_CleanPrice
PriceAccrued	MarkitPV_PriceAccrued
PV01Local	MarkitPV_PV01Local
PV01Local	MarkitPV_ PV01Local
FAS157Rating	MarkitPV_FAS157Rating

3.2.8 Configuring Results - Pricer Measures

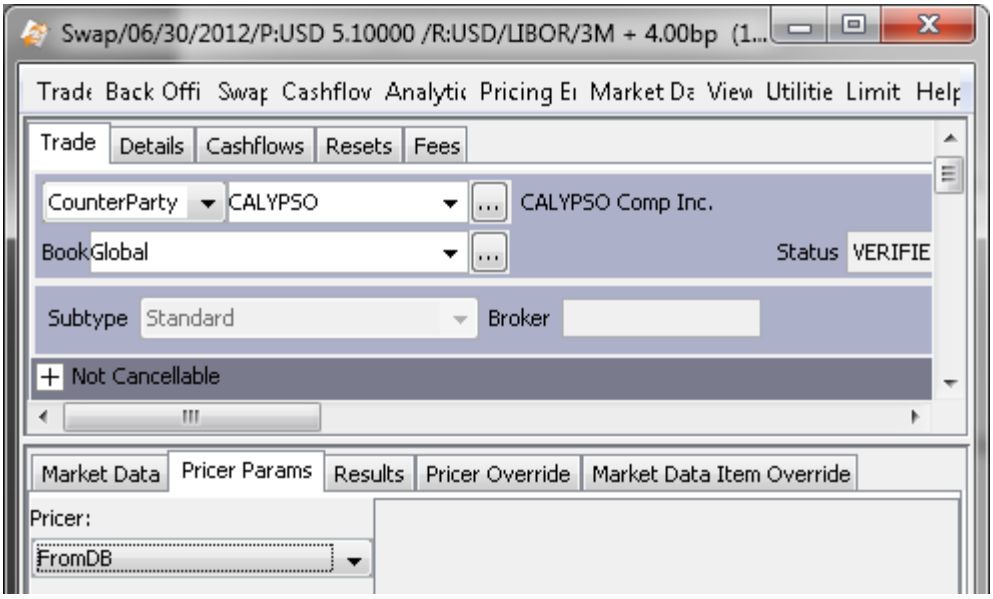
Choose **Trade > Configure Results** to select the pricer measures you want to display in the Trade Window.



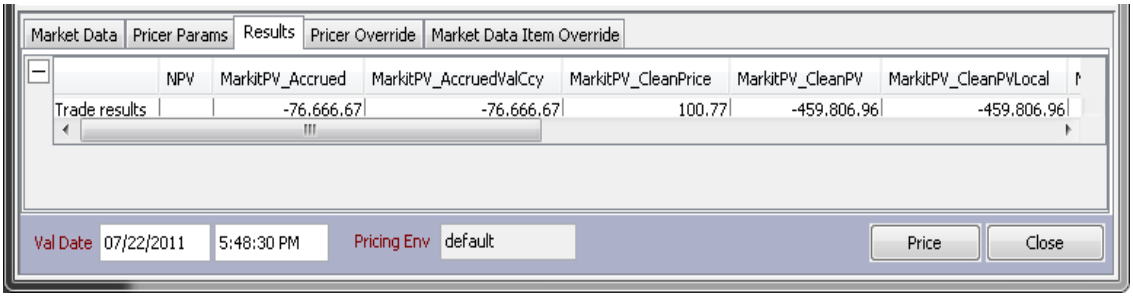
After completion of the import scheduled task, the Markit valuations are viewable at the trade level.

PricerFromDB

It only shows the custom measures as shown in the Trade GUI below:

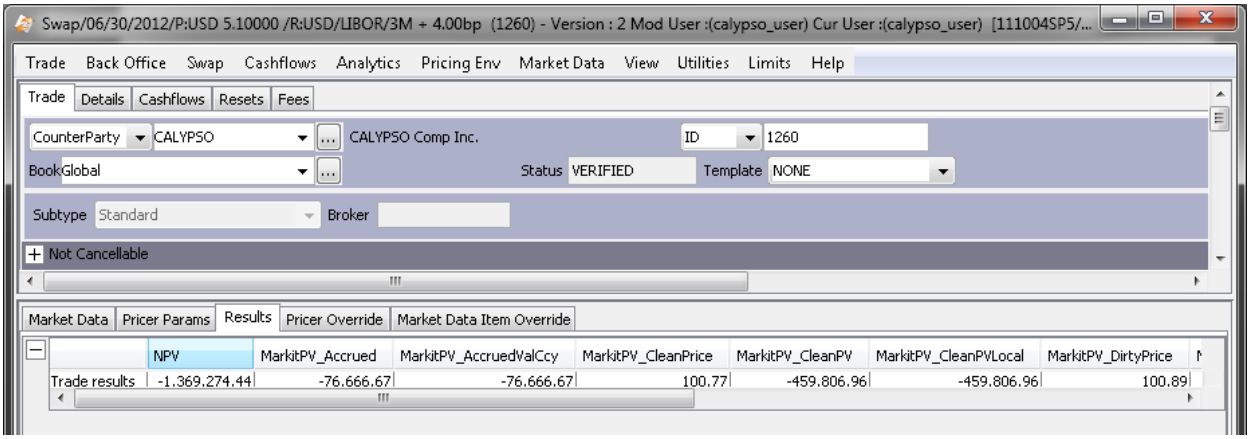


You can view the measure values in the “Results” tab after clicking “Price”.



Trade Price

It is also possible to view both the custom measures as well as the calypso measures side by side. The pricer to be selected in that case will be trade specific. As shown in below figure.



You can view both the calypso measure “NPV” as well as the custom markit measures like “MarkiPV_Accrued” etc.

It is also possible to retrieve the Markit Valuations stored in the database for any specific date using the PricerFromDB report: