CALYPSO

Calypso Data Uploader

Calypso Data Uploader — Integration Guide

©1997—2011, Calypso Technology Inc.

Document History

Revision date	Edition number	Comment
25 June 2010	First Draft	0.1
28 June 2010	Second Draft	0.2
30 June 2010	First Edition	1.0
23 July 2010	Second Edition	1.0.1
03 September 2010	Third Edition	1.0.2
30 September 2010	Fourth Edition	1.0.3
31 March 2011	Fifth Edition	1.0.4

Contents

Figures

Changes

Rel 1.3 — 29 April 2011

Data Uploader Service

The Data Uploader is now responsible for making data connections. Users must configure the DataUploaderService in `service.properties`. See Step 6 on page 14 for complete information.

cal-uploader moved to DataUploader Release Jar

The cal-uploader is no longer a separate download. It is distributed with the DataUploader release jar.

FRA Rounding

Users can now set the rounding to use when uploading FRA deals. See Section 4.4.1, "FRA Rounding," on page 23.

ForceAmend

It is possible to force an amend on a trade with the message workflow rule, **ForceAmend** for instances when interface is down and a trade is entered using a manual link to an existing MarkitWire trade in Calypso. See Section 4.4.1, "FRA Rounding," on page 23 for further information.

Copp Clark Holidays

The Data Uploader now supports uploading Copp Clark Holiday files. See Section 4.4.3, "Copp Clark Holidays," on page 24 for compelte information.

New Products

- FX Spot
- FX Forward
- Future Equity
- Future Equity Index
- Equity Swap
- Swaption
- TRS
- Variance Swap
- ZC Inflation

1   Calypso Data Uploader

The Calypso Uploaders provide support for Calypso 12.0. Each Calypso base release has its own uploader. Refer to Table 3-1, "Calypso Data Uploader Version Compatibility," on page 13 to determine which versions of the Data Uploader and cal-upload file are compatible with your Calypso implementation.

Note: The documentation for each Calypso Integration package that makes use of the Data Uploader will also contain its own compatibility table.

1.1   Introduction

The Calypso Data Upload Framework (CDUF) simplifies importing data into Calypso. It supports many product types as well as certain reference data.

CDUF supports the all trade life cycle events:

- New
- Amend
- Terminate (Partial and Full Termination)
- Novate (PO, CounterParty, and Incoming Party. Partial and Full Novation)
- Cancel
- Undo Terminate
- Exercise

In addition to Trades, CDUF also supports uploading

- Books
- Legal Entities
- Corporate Actions

1.2   Features

- Full Lifecycle support — This supports the full lifecycle for trades – from New to Terminates. It also support custom workflow actions defined in workflows.
- Calypso BO Message-based framework — This allows for extensibility by allow customers to add their own validations via workflow rules. The workflow can also be customized to meet customer needs.
- Trade Lifecycle Ordering — The trade lifecycle actions are played back in the same order as they are received. If one of the earlier actions cannot be applied due to a problem, all subsequent actions on the trade are blocked as well.

- Error Reporting — All errors are available in the Task Station for review by your operations staff, or by Calypso operations staff on the SaaS implementations.
- Replay Messages — Trades in errors are blocked as a message and can be played back when the underlying system error is corrected.
- Native XML format — The uploader has a native XML format. It supports multiple type of Keywords and Fees.
- Acknowledgement Framework — In addition to Task Station integration, there is a publisher framework to interface with electronic Trade Capture systems, such as MarkitWire, to send acknowledgements and trade capture status to these systems.
- Calypso Termination API — The uploader uses the Calypso Termination Extensions. As a result, it supports any extensions created by users and allows for configurable actions for Termination and Novation.
- Validation Framework — All validations are done via Calypso BO Message Workflow rules. This allows a single code-base to be used for Validations and Actions across different incoming formats. For example, the uploader supports inputs from MarkitWire, ICELink, CSV, as well as the Calypso Upload Document format (XML). All business validations are performed in one place.

  The required effort to interface with a new trade capture is reduced to translating the new format to the Calypso Upload Document format (assuming the native XML supports those features).
- Extensions — The Framework supports extensions as follows:
  - New workflow rules to handle customer-specific business rules.
  - Implementation teams can supply their own CSV formats convertors to create CSV upload files.

1.3   Input Sources

The following input sources are currently supported by this Calypso Uploader Framework

- **XML** — Standardized Calypso Upload XML, also known as the Calypso Upload Document[1] (CUD) format.
- **MarkitWire** — A new MarkitWire interface for IRDSwaps trades. Refer to the Calypso MarkitWire Integration Guide.
- **ICELink** — Credit Derivatives Affirmation Interfaces. (Forthcoming)
- **CSV** — Support for limited products in CSV format. Extendable by the implementation team.

---

1. Calypso Upload Document Format differs from Calypso ML format.

1.4   Supported Products

- **Interest Rate Default Trades** —Vanilla Swaps, Caps, Floors, XCCY Swaps, Basis Swaps, CMS, Swaption, Cancelable Swap
- **Credit Rate Default Trades** — CDS, CDS Index Tranche, Loan CDS, Recovery Swap, CDX, ABS Index, TRS-Bond, TRS-Indices.
- **Equity Derivatives Trades** — Single, Index, OTC Equity Options, Variance Swap.
- **Fixed Income** — Fixed Income Security Trades (Corporate, Govt / Treasury Bonds, etc)
- **Equity** — Equity Cash.
- **Miscellaneous** — Books, Legal Entities, Legal Entity Contacts, Corporate Actions (for Equity only), Users, User Groups.

Table 1-1 lists the supported Product Types and Product Sub-types for XML-format data.

Table 1-1: Product Type/Sub-type Support Matrix

Product	Product Type	Product Sub-type
CreditDefaultSwap- SingleName	CreditDefaultSwap	
CreditDefaultSwapLoan	CreditDefaultSwap	CreditDefaultSwapLoan
CDSIndex	CDSIndex	
CDSABSIndex	CDSABSIndex	
Recovery Swap	CreditDefaultSwap	RecoverySwap
InterestRateSwap Vanilla	InterestRateSwap	
InterestRateSwap OIS	InterestRateSwap	OISSwap
InterestRateSwap XCCY	InterestRateSwap	CrossCurrencySwap
InterestRateSwap BasisSwap	InterestRateSwap	BasisSwap
SimpleTransfer	SimpleTransfer	
ETO	ETO	
Equity	Equity	
Bond	Bond	
FX Spot	FX	FXSpot
FX Forward	FX	FXForward
Future Equity	Future	FutureEquity
Future Equity Index	Future	FutureEquityIndex
EquitySwap	EquitySwap	
CapFloor	CapFloor	
Swaption	Swaption	
EquityStructuredOption	EquityStructuredOption	
FRA	FRA	
OTC Equity Option	OTCEquityOption	

Table 1-1: Product Type/Sub-type Support Matrix (Continued)

Product	Product Type	Product Sub-type
TRS	TotalReturnSwap	
VarianceSwap	VarianceSwap	
ZC Inflation	InterestRateSwap	ZCInflationSwap

2 Installation and Setup

2.1 Operation Overview

The Data Uploader is invoked either by the File Watcher application or by a Calypso Engine.

The File Watcher monitors a specified folder at a specified interval, and when a new file is found, invokes the configured uploader.

Note: The "watched" file directory and the interval between checks is user-configurable.

Input files must be comma separated values (.CSV) or in the Calypso Upload Document (CUD) XML format[1]. For CSV- and XML-formatted file, the file name must start with the product name. For example, the name of a CSV or XML file containing FX trades must begin with FX: `FX_20100226.csv`, `FX_1Q2010.xml`, etc. There are no restrictions on the remaining characters of the filename.

The file extension indicates the format of the contents, CSV for a file containing data in comma separated values format, or for example, XML for a file containing data in the Calypso Upload Document format.

When a CSV file is loaded, the application converts it to the CUD using XSLT Transformation. The XML files are next processed into BO Messages, where they are validated before being processed into Trades and uploaded to the database.

For trade data, the **External Ref ID** field must be unique. The **Bundle Type** and **Bundle Name** are generated on the fly. For example, in a file of fills for one order on an ECN, the **Bundle Name** can be the order number; the **External Ref ID** will represent the **ECN Trade ID**.

Note: On the Calypso SaaS platform, the only supported file type is CSV.

2.1.1 Manual Uploading

To manually upload trades, launch the FileWatcher application and copy an input file into the folder specified in the configuration file. The FileWatcher invokes the Calypso Data Uploader at the end of the current wait interval.

The Data Uploader processes the input file and creates an XML result file having the same base filename (e.g., `FX_1Q2010`) in the same directory. The result file contains details on the uploads, and depending on the outcome of that processing, either has a `.finished` or `.failure` file extension, to indicate success or failure, respectively.

---

1. Example Calypso Upload Documents for each product type are available in `$CALYPSO_HOME\resources`.

Note: Certain Calypso engines internally process incoming data into CUD format, making it ready for upload to the database without needing external files.

2.1.2   Data Handling for Life Cycle Events

Table 2-1: Data Handling for Life Cycle Events

Lifecycle Event	Trade Data	Keyword Action	Fees Action
NEW	Save All	Save All	Save All
AMEND	OVERWRITE	Overwrite Existing Keywords. Add New Keywords.	Remove existing Fees and replace them from XML
CANCEL	Perform Basic Validation, do not Overwrite other trade data.	Overwrite Existing Keywords. Add New Keywords.	Keep existing Fees and add new fees from XML
TERMINATE	Perform Basic Validation, do not Overwrite other trade data. Refer to the TERMINATION block for termination tags to perform the Termination.	Overwrite Existing Keywords. Add New Keywords.	Keep existing Fees and add new fees from XML
PARTIAL TERMINATE	Perform Basic Validation, do not Overwrite other trade data. Refer to the TERMINATION block for termination tags to perform the Partial Termination.	Overwrite Existing Keywords. Add New Keywords.	Keep existing Fees and add new fees from XML
NOVATE	Perform Basic Validation, do not Overwrite other trade data. Refer to the NOVATION block for novation tags to perform the Novation.	Overwrite Existing Keywords. Add New Keywords.	Keep existing Fees and add new fees from XML

Table 2-1: Data Handling for Life Cycle Events (Continued)

Lifecycle Event	Trade Data	Keyword Action	Fees Action
PARTIAL NOVATE	Perform Basic Validation, do not Overwrite other trade data. Refer to the NOVATION block for nova- tion tags to perform the Partial Novation.	Overwrite Existing Keywords. Add New Keywords.	Keep existing Fees and add new fees from XML
UNDO TERMINATE	Perform Basic Validation, do not Overwrite other trade data.	Overwrite Existing Keywords. Add New Keywords.	Keep existing Fees. Ignore Fees from XML. Remove TERMINATION_FEES
Basic Validation	Validate that the book and counterparty of the Action matches those of the existing trade.		

2.1.3   General Notes

1. Adding an (non-unique) index for the External Reference column of the Trade Table can improve performance.

2. Adding an index on the Attribute Type + Attribute Value columns of the Message Attribute table can improve performance.

2.2   Known Issues

- When a Basis Swap is Partially Terminated in Calypso, the first Rate for the newly-created residual trade (i.e., the child trade) is cleared. This situation also applies to Cash Settlement information for Cap/ Floor trades.

  This is a core Calypso issue and not a problem specific other inter- faces.

- When using the German locale, **Init Level of Inflation Swaps** is stored incorrectly. If this issue impacts your implementation, please initiate a Calypso HelpDesk issue and request a hotfix for CAL- 97335. This is issue was resolved in 11.1.0.3 Calypso.

3   Setup Instructions

3.1   General Setup

Use the following procedure to install the Data Uploader.

Use the following table to determine which Data Uploader and cal-uploader versions are required for your Calypso Release:

Table 3-1: Calypso Data Uploader Version Compatibility

Base Calypso Release	Module Name	Required Module Version	Module dependency (in sequence)
11.1.0.X	cal-upload	1.2.4	Calypso 11.1.0.1+
	datauploader	1.0.3	Calypso 11.1.0.1+, cal-upload 1.2.1
11.0.0.2	cal-upload	1.2.4	Calypso 11.0.0.2
	datauploader	1.0.3	Calypso 11.0.0.2, cal-upload 1.1.2

Note: Substitute the correct version numbering for *x.x*.

1. Download the compressed deliverable file (`datauploader-x.x.x-rel.jar`) from the Calypso download site.

2. Extract the zip file into `$CALYPSO_HOME`:

   `jar xvf datauploader-x.x.x-rel.jar`

   The deliverable contains the following directories and files:
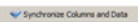
```
bin/dbscripts/GatewaySchemaBase.xml
bin/dbscripts/GatewaySchemaData.xml
docs/data_uploader.pdf
jars/jdom-x.x.jar
jars/datauploader-x.x.jar
jars/edtftpj-pro-3.1.2.jar
jars/cal-upload-x.x.jar
resources/datauploader.properties
resources/edltlicense.properties
resources/ErrorCodeBundle.properties
resources/GATEWAYMSG.wf
resources/gatewayservice.properties
resources/sftpuploader.properties
resources/uploadcsvmapping.properties

resources/calypsox/mapping/BondCSVMapToCalypsoUploadDocument.xslt
resources/calypsox/mapping/IRSCSVMapToCalypsoUploadDocument.xslt
sql/GatewaySchemaData.xml
```

3. Set the classpath in `$CALYPSO_HOME/bin/calypso.bat or .sh`
   a. Add the following lines to the "set CLASSPATH" section, subs ti tu i ng the correct version number for *x.x*:

      `%CALYPSO_HOME%\cal-upload-x.x.jar;^`

      `%CALYPSO_HOME%\datauploader-x.x.jar;^`

   b. Add the following line to the "set CALYPSO_LIB" section, substituting the correct version number for *x.x*:

```
%CALYPSO_LIB%\jdom-x.x.jar;^
```

The version of jdom shipped with the Data Uploader is newer that the version that shipped with Calypso 11.0.x–11.1.x. **You must also add this version to the CLASSPATH**.

   c. `edtftpj-pro-3.1.2.jar` is not used at this time and need not be added to the classpath.

4. Run ExecuteSQL to update the database with the required domain values.

   a. Remove `bin/dbscripts/SchemaData.xml`.

   b. Add `$CALYPSO_HOME/sql/GatewaySchemaData.xml`.

   c. Add `$CALYPSO_HOME/sql/GatewaySchemaBase.xml`

   d. Select the **gateway** check box.

   e. Click ![Synchronize Columns and Data]. ExecuteSQL will generate the SQL to synchronize your database.

   f. ExecuteSQL displays a message dialog when the SQL is ready. Click **Okay** to continue.

   g. Click ![►] to synchronize your database.

5. (optional) Modify `resources/gatewayservice.properties`.

The application converts imported data into BOMessages, which require a Legal Entity for the Sender and Receiver.

By default, `gatewayservice.properties` sets the Sender to CLIENT and the RECEIVER to CALYPSO. If these entities are not present as Legal Entities, Calypso will create them.

To use a different Sender (i.e., Client) or Receiver (Host), modify the following lines in the `gatewayservice.properties` as desired:

```
GatewayServiceClientName=CLIENT
GatwayServiceHostName=CALYPSO
```

6. Modify `$CALYPSO_HOME\resources\service.properties`.

   a. Add an entry for **DataUploadServer** to the services line.

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

   b. Add the following lines. **Make each of these lines a single-line statement in the file**.

```
DataUploadServer.name=DataUploadServer

DataUploadServer.class=com.calypso.tk.service.DataUploadServerImpl

DataUploadServer.port=0

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

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

Note: Refer to in the *Calypso Installation and Upgrade Guide* for further information on the `service.properties` file, including instructions for dealing with firewalls. There is no need to modify the information except for firewall requirements.

7. Launch the Data Server and Main Entry.

3.2 Workflow Configuration

1. Launch the Workflow Config window (**MainEntry->Configuration > Workflow > Workflow...**) and import (**Workflow Config-> File > Import All**) `GATEWAYMSG.wf` into Calypso. `GATEWAYMSG.wf` provides the standard workflow for Gateway BO Messages.

   `GATEWAYMSG.wf` is a *Message* workflow not a Trade workflow. Therefore, before importing, choose:

   - EventClass: **PSEventMessage**
   - Subtype: **GATEWAYMSG**
   - Product: **ALL**

Warning: Ensure that the settings above are used or existing workflows will be overwritten.

Before importing, your Workflow Config window should appear similar to :

Figure 3-1: Workflow Config Window — Preparing to Import GATEWAYMSG.wf
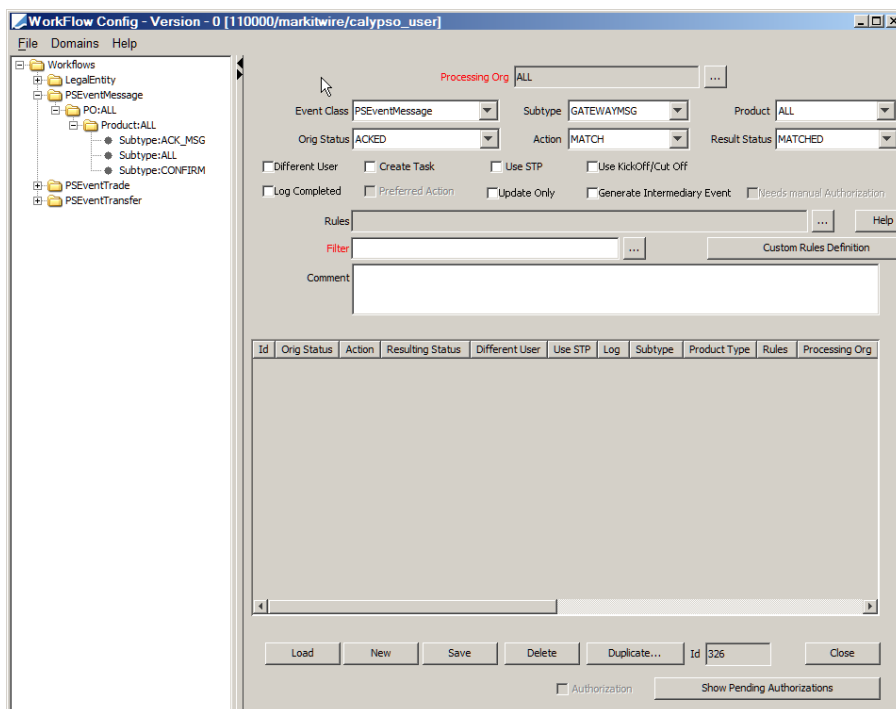
2. Select **File > Import All...**, then use the file browser to navigate to `$CALYPSO_HOME/resources/GATEWAYMSG.wf`.

   After selecting `GATEWAYMSG.wf` in the file browser, click **Open**. The application imports the workflows and then displays a confirmation dialog showing what actions were accomplished:

Figure 3-2: Workflow Import Confirmation
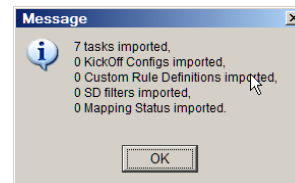
The Gateway Message workflow is now available:

Figure 3-3: Gateway Workflow

| Id	Orig Status	Action	Resulting Status	Different User	Use STP	Log	Subtype	Product Type	Rules	Processing Org	Kick Off/Cut
1001	NONE	NEW	PENDING_VALID	☐	☑	☐	GATEWAYMSG	ALL		ALL	☐
1002	PENDING_TRADE	CANCEL	CANCELED	☐	☐	☐	GATEWAYMSG	ALL	CancelCleanUp	ALL	☐
1003	PENDING_TRADE	LOAD	COMPLETED	☐	☑	☐	GATEWAYMSG	ALL	CheckLink,Loader	ALL	☐
1004	PENDING_TRADE	REPROCESS	PENDING_TRADE	☐	☐	☐	GATEWAYMSG	ALL	ReMap	ALL	☐
1005	PENDING_VALID	CANCEL	CANCELED	☐	☐	☐	GATEWAYMSG	ALL	CancelCleanUp	ALL	☐
1006	PENDING_VALID	REPROCESS	PENDING_VALID	☐	☐	☐	GATEWAYMSG	ALL	ReMap	ALL	☐
1007	PENDING_VALID	VALIDATE	PENDING_TRADE	☐	☑	☐	GATEWAYMSG	ALL	CheckLink,Validate	ALL	☐

Figure 3-4: Workflow Graph — GATEWAYMSG Graph
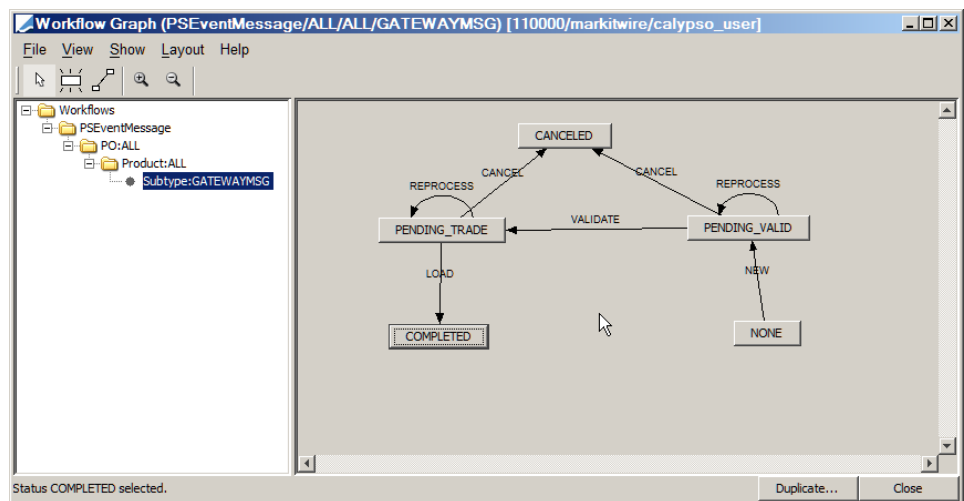
3. Trade Workflow setup.

   Termination, Novation, and Cancellation Trade workflow rules must be set.

a. For the **Verified -> Termination** Transition, setup the UpdateTermination Workflow rule.

Figure 3-5: Workflow Config — UpdateTermination
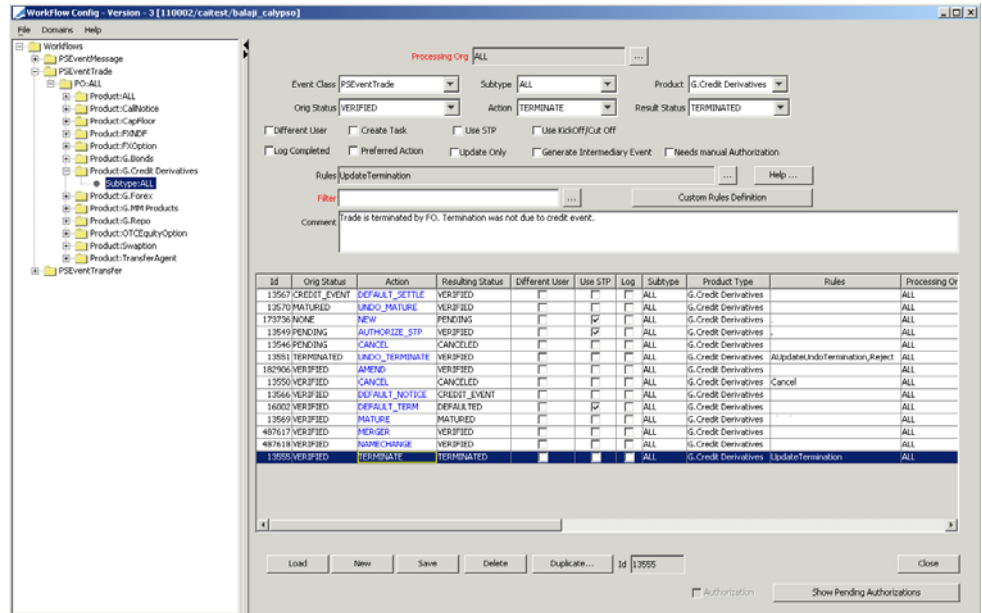
b. For **Cancel** action, setup Cancel.
c. For **Undo_Termination**, setup **AUpdateUndoTermination** and **Reject**. Note that **Reject** is core Calypso workflow rule.

Figure 3-6: Workflow Config — AUpdateUndoTermination and Reject
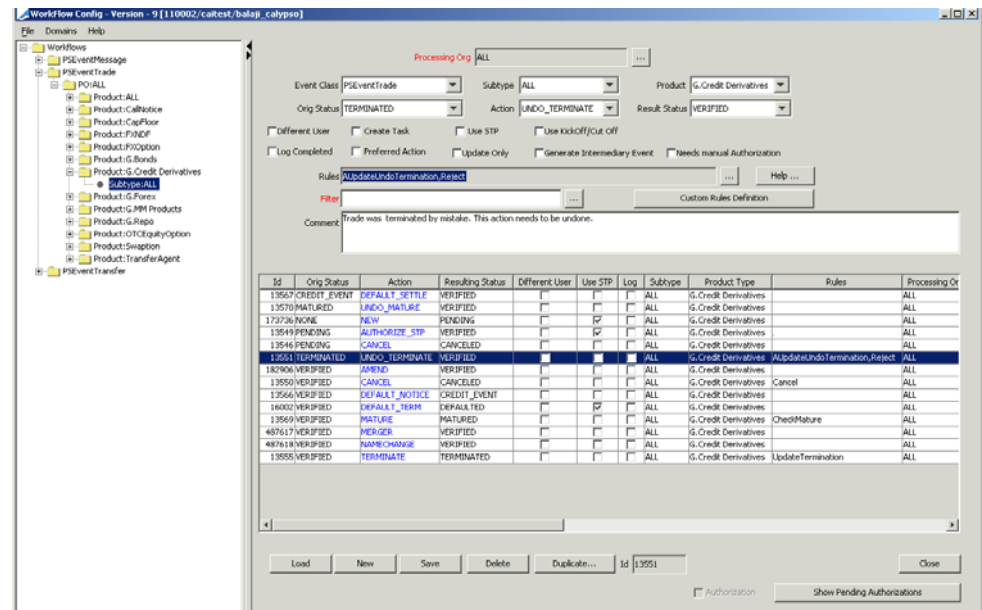
    d. The remaining rules are specific to the Data Uploader and are mandatory.

Note: Any changes to the BO Message workflow rules must be reflected in the `gatewayservice.properties` to indicate which states are for blocking messages. These are used to link together the BO Messages.

4. Verify that the GATEWAY exceptionType is present in Domain Values. If it is not present, add it now:

Note: If Main Entry was active when you ran ExecuteSQL, you will need to restart Main Entry before any changes appear.

Figure 3-7: Domain Values — GATEWAY exceptionType
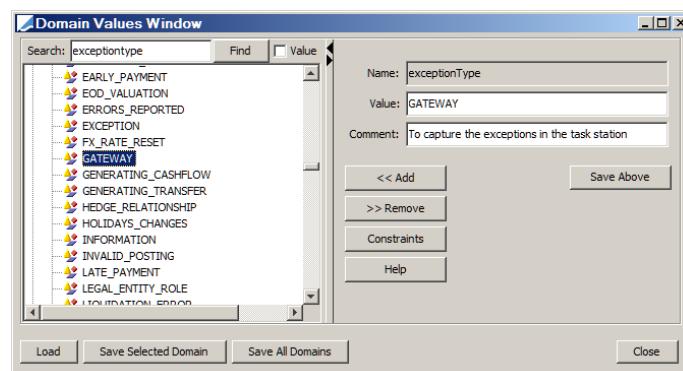
Note: Prior to processing Terminations, Novations and Cancellations, you must have created the supporting Trade Workflow rules.

5. Define the following Domain Values to support Trade Workflow rules:

    a. **UploadTerminateAction**:

    This key holds the action to apply when a trade is terminated. If this value is missing, the default is TERMINATE.

    b. **UploadNovateAction**:

    This is the workflow action to apply when a trade is novated. If the value is missing, the default is TERMINATE.

    c. **UploadExitAction**:

    This is the workflow action to apply when you must exit from a outside trade (e.g., a MarkitWire trade). This is action will affect the upstream trade outside Calypso.

    d. **UploadTerminationReason**:

    This provides the default reason for a trade termination. The reason is stored as a keyword on the trade. If this value is missing, the default is Assigned.

    e. **UploadNovationReason**:

      This provides the default reason for a partial trade termination. The reason is stored as a keyword on the trade. If this value is missing, the default is Assigned.

6. Define a new rule, ForceNew under the GATEWAYMSG workflow:

Figure 3-8: GATEWAYMSG Workflow — FORCE_NEW Action
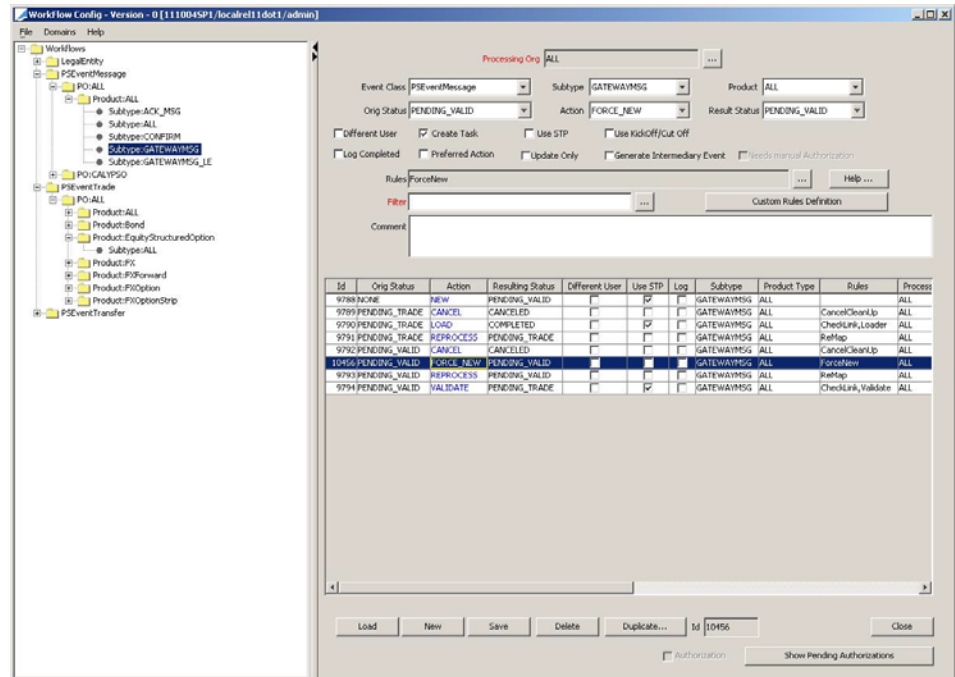
3.3 Task Station Setup

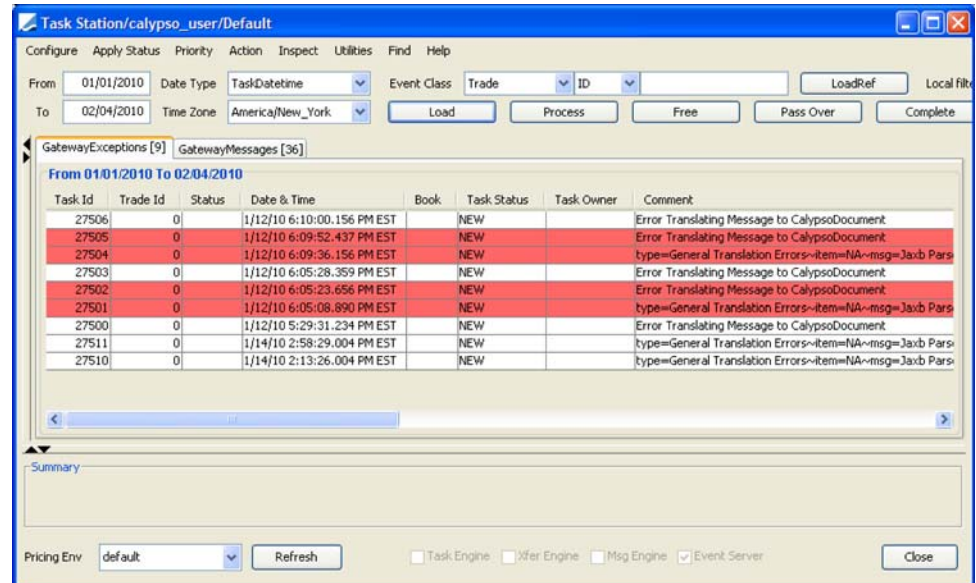1. Launch the Task Station (**Processing > Task Station...**)

2. Launch User Task Station Defaults (**Task Station -> Configure >
    User Config...**). Add tabs for Gateway Messages and Exceptions to
    the Task Station:

Figure 3-9: User Task Station Defaults

When Gateway exceptions occur, you will then be able to see them in the Task Station:

Figure 3-10: Task Station — Gateway Exceptions

3.4   Configure and Launch the FileWatcher

Note: This section is only required if you will be performing manual uploads of CSV or XML files. It is not a requirement to use Calypso Engines, such as the MarkitWire engine.

3.4.1   Configuration Files

The distribution jar contains example configuration files for multiple data formats. Use the configuration file that matches the data format of the data you wish to upload.

The sample `datauploader.properties` file appears similar to the following example:

```
# This is the encoding scheme to be used for the INPUT (i.e. CSV) files
# This feature is provided to take care of CSV files created using different
# encoding schemems e.g. SHIFT_JIS for Japanese
INPUT_ENCODING_SCHEME=UTF-8

# When GENERATE_FILES is true, the Data Uploader places the intermediary files in the
# CALYPSO_HOME/DataUploader folder when creating CSV file (XML before XSLT and
# after applying the XSLT)

GENERATE_FILES=true

# This is the encoding scheme for the output of the CSV uploader.
# The CSV uploader creates an XML file that is read by the XML uploader
# The output scheme instructs the CSV uploader to create the output XML from
```

```
# the CSV uploader
# (input for the XML uploader) in this encoding scheme
OUTPUT_ENCODING_SCHEME=UTF-8

# File Watcher Properties
#This is the folder where we place the files.
fileDir=c:\\uploader\\
# Look for file changes every xxx milli-seconds.
interval=1000

# This tells the file watcher to call the XML Uploader
fileWatcherType=data

#This tells the file watcher to monitor CSV & XML Files only
fileFilter=csv,xml
```

3.4.2   Launching FileWatcher

Use a version of the following command line to launch the FileWatcher from a terminal or from script based on a script from the `$CALYPSO_HOME/bin/`*OS_Platform*`/jdk` directory:

```
java -Xmx256m -D com.calypso.apps.startup.StartFileWatcher
-propertyFile datauploader.properties
-env YourEnv -user user_name -password xxxxxxx
```

Based on your volume, you may need to increase the memory allocated to the FileWatcher (e.g., `-Xmx384`, `-Xmx512`). With the FileWatcher operating, you can then place a file to import in the watched directory (specified by `fileDir`). At the end of the current wait interval, the FileWatcher will notify the Data Uploader which will then process the file accordingly.

The Task Station will display any errors that may occur.

4   Operation

4.1   Requirements

Before using the Data Uploader, the Data Server, Event Server, and Task Engine must be running. Additionally, you must monitor the Task Station for errors and blocked messages.

4.2   Handling Blocked Messages

In the event of a blocked message, after clearing the issue, you can use the VALIDATE or LOAD action to re-process a manually uploaded message. For messages received from an engine, use the action appropriate for that system.

For example, after clearing an issue for a blocked MarkitWire message, use the REPROCESS action to resubmit the message to the system.

For messages from external systems, refer to the specific Calypso integration guide for that system.

4.3   Task Station Errors

If Task Station error messages are improperly displayed, ensure that `ErrorBundle.properties` is present in the resources directory.

4.4   Misc Data Uploader Changes

4.4.1   FRA Rounding

You can set the default rounding method for uploaded FRA data by setting the **UploadDefaultRoundingMethod** domain value to UP, DOWN, or NEAREST. By default, uploaded data is rounded to Nearest.

4.4.2   ForceAmend

It is possible to force an amend on a trade with the message workflow rule, **ForceAmend** for instances when interface is down and a trade is entered using a manual link to an existing MarkitWire trade in Calypso.

Use the Domain Values window to add the **ForceAmend** message rule under the **workflowRuleMessage** domain and the FORCE_AMEND action under the **messageAction** domain.

Configure the FORCE_AMEND Action and **ForceAmend** rule, between the original and the result status = PENDING_VALID of GatewayMsg workflow:

Figure 4-1: Configuring the FORCE_AMEND Action
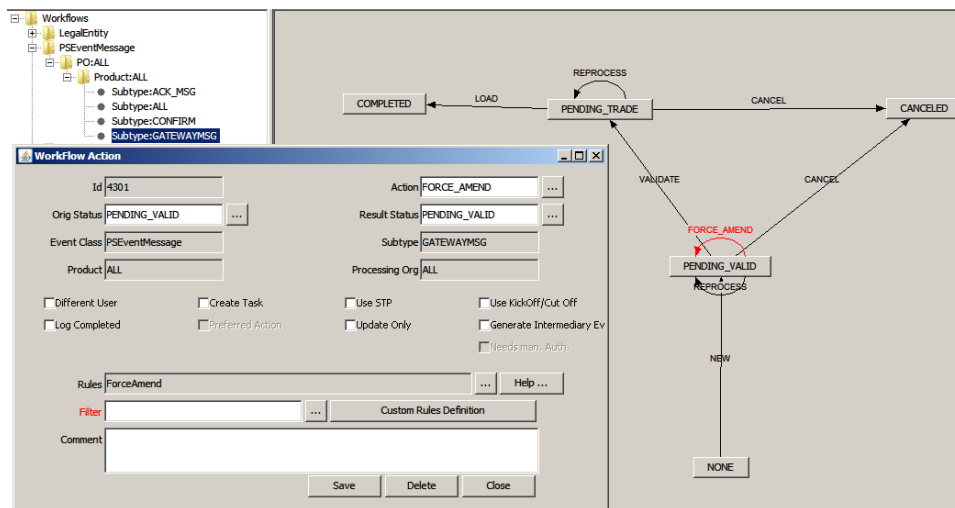
If a trade with given External Reference already exists in calypso, then a new MarkitWire trade won't be uploaded to Calypso. Such a trade will fail with exception task "Trade Already exists with External Reference :

MW_CALYPXXXX_1234567" and the GatewayMsg in PENDING_VALID state.

To replace the existing trade with the new MarkitWire trade, manually apply the FORCE_AMEND action on the gateway message. The action in Calypso trade be forced from NEW to the default Amend Action and the MarkitWire trade will then be uploaded into Calypso with the Amend Action.

4.4.3   Copp Clark Holidays

The Data Uploader supports the Copp Clark interface for importing Holidays into Calypso.

Note: Copp Clark support requires that your implementation be based on a Calypso Fast-Track database. Contact your Calypso Account Manager for further information.

Copp Clark provides a tab-delimited file with Calypso Fast Track holiday codes.

Copp Clark data is imported via the ScheduledTaskCOPPCLARK_FEED. Scheduled Task. The Scheduled Task has the following attributes:

- TYPE: Import
- Upload: Uploads the downloaded Copp Clark file from the Output Location to Calypso.
- DeleteEventDates: Yes / No

  If Yes, the application deletes all EventDates in all the HolidayCodes that are between **Start Date** and **End Date**. Note that Copp Clark provides all events from Today through 30–40 years. This necessitates deleting existing data.

  If No, the application does not delete any events, but simply adds to the existing list. Adding does *not* create duplicate events, nor will it remove any future holiday events that were previously setup in error.

- Delete Start Date: Date in YYYYMMDD format
- Delete End Date: Date in YYYYMMDD format
- Input file: File Name (Single File Name or Comma separated list of files)

  If a filename is specified, the application will upload the specified file (or a comma-separated list of files) from the Input Location. .

  If no filename is specified, the application will upload all available text files in the temp/Output Location.

5   Publisher Engine

Generally, publishing the acknowledgement to the sender is handled by the Acknowledgement Publisher classes which are called by the Gateway Processor.

However, if for some reason the inital trade message is blocked and later processed by the user, the acknowledgement is then sent by the UploaderPublishEngine rather than the Acknowledgement Publisher classes.

The administrator must configure the **UploaderPublishEngine** just as with other engines.

Step 1 — Add the Engine ID for the UploaderPublishEngine:

Configure the **UploaderPublishEngine** with an Engine ID:

Figure 5-1: Engine Configuration — UploaderPublishEngine

Step 2 — Configure the Event Consumer:

Add the **UploaderPublishEngine** as the consumer for the
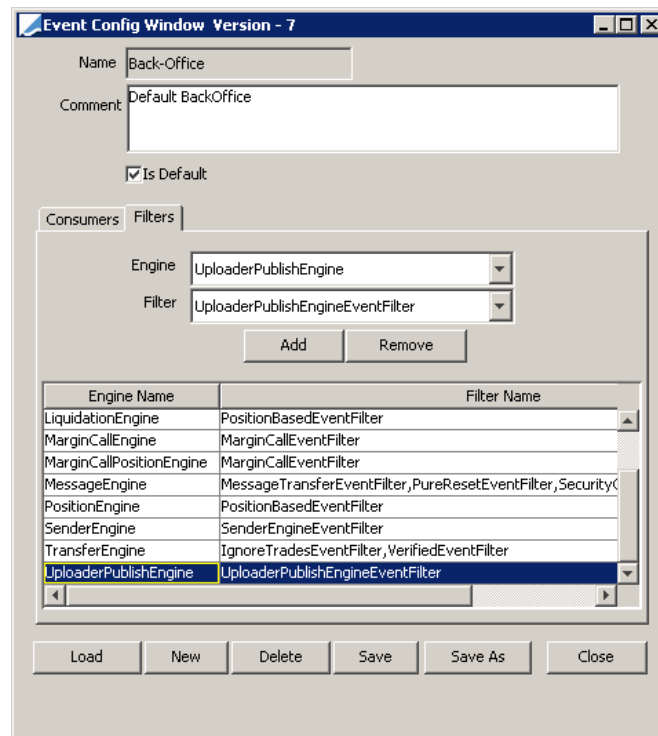**PSEventRepublish** Event Class:

Figure 5-2: PSEventRepublish Event Class Configuration
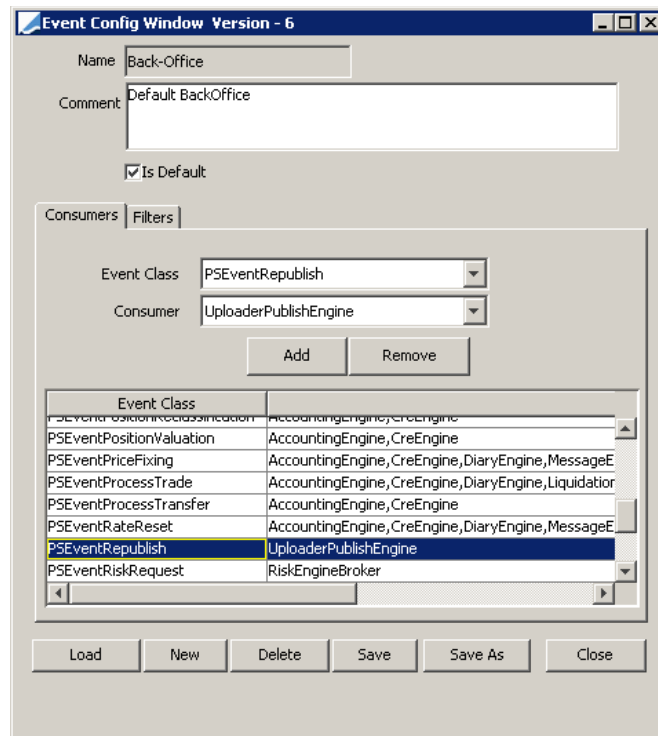
Step 3 — Configure the Event Filter for the UploaderPublishEngine

Configure the UploadPublishEvent Filter Name

Figure 5-3: Configuring the UploadPublishEngine Event Filter

Contacting Calypso

Americas

SAN FRANCISCO

595 Market Street
Suite 1800
San Francisco, CA 94105
USA
T   +1 415 817 2400
F   +1 415 284 1222
info@calypso.com

NEW YORK

99 Park Avenue,
Suite 930
New York, NY 10016
USA
T   +1 212 905 0700
F   +1 212 905 0724
sales_nyc@calypso.com

Europe, Middle East, Africa

LONDON

17 Dominion Street
London EC2M 2EF
United Kingdom
T   +44 20 7826 2500
F   +44 20 7826 2501
sales_london@calypso.com

PARIS

106/108 rue de La Boëtie
75008 Paris
France
T   +33 1 44 50 13 99
F   +33 1 44 50 12 84
sales_paris@calypso.com

FRANKFURT

Junghofstrasse 24
2nd floor
60313 Frankfurt Am Main
Germany
T   +49 69 920389 0
F   +49 69 920389 11
sales_frankfurt@calypso.com

JOHANNESBURG

Nelson Mandela Square
2nd Floor, West Tower
Maude Street
Sandton, 2196
South Africa
T   +27 11 881 5708
F   +27 11 881 5611
sales_johannesburg@calypso.com

COPENHAGEN

Calypso Technology Denmark
Regus City Center, Office 308
Larsbjornstraede 3,1454 Copenhagen
T   +45 33 37 71 79
F   +45 33 32 43 70
sales_copenhagen@calypso.com

MOSCOW

Office number 532
4, 4th Lesnoy pereulok
125047 Moscow Russia
T +7 495 663 80 63
F +7 495 225 8500
sales_moscow@calypso.com

Asia Pacific

TOKYO

Kojimachi Place 10F
2-3 Kojimachi
Chiyoda-ku
Tokyo 102-0083
Japan
T   +81 (0)3 5214 1800
F   +81 (0)3 5214 1801
sales_tokyo@calypso.com

SINGAPORE

1 Phillip Street, #12-02
Singapore 048692
Singapore
T   +65 6372 1121
F   +65 6372 1161
sales_singapore@calypso.com

SYDNEY

Level 4, 95 Pitt Street
Sydney, NSW 2000
Australia
T   +61 2 8249 8115
F   +61 2 8249 8116
sales_sydney@calypso.com

MUMBAI

Unit 401, Akruti Center Point
MIDC Central Road
Near Marol Tel. Exchange
MIDC, Andheri (E)
Mumbai 400 093
India
T   +91 22 6681 5600
F   +91 22 6681 5611
sales_mumbai@calypso.com

CHENNAI

Plot No.23 & 24
Door No.33
South Beach Avenue
MRC Nagar, Chennai
Tamil Nadu 600028
India
T   +91 44 4347 5350
F   +91 44 4347 5378
saleschennai@calypso.com

HONG KONG

Unit 32, Level 21, The Center
99 Queen's Road, Central
Hong Kong
T   +852 3478 3796
    +852 3478 3797
F   +852 3478 3880
sales_hongkong@calypso.com