

Nasdaq Calypso Bloomberg FIT Integration Guide (Generic)

Version 4.12.1

Revision 16.0 June 2023 Approved



Copyright © 2025, Nasdaq, Inc. All rights reserved.

All content in this document is owned, or licensed, by Nasdaq, Inc. or its affiliates ('Nasdaq'). Unauthorized use is prohibited without written permission of Nasdaq.

While reasonable efforts have been made to ensure that the contents of this document are accurate, the document is provided strictly "as is", and no warranties of accuracy are given concerning the contents of the information contained in this document, including any warranty that the document will be kept up to date. Nasdaq reserves the right to change details in this document without notice. To the extent permitted by law no liability (including liability to any person by reason of negligence) will be accepted by Nasdaq or its employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

Document History

Revision	Published	Summary of Changes
1.0	May 2016	First edition.
2.0	July 2016	Second edition – Updates for supported Calypso version.
3.0	May 2017	Third edition – Added Trader mapping.
4.0	May 2017	Fourth edition – Added SSL Connection.
5.0	June 2017	Fifth edition – Added TLS Connection.
6.0	September 2017	Sixth edition – Updates for supported Data Uploader version. Added Proxy Support.
7.0	March 2018	Seventh edition – Added 15.X installation.
8.0	November 2018	Eighth edition – Updates for Calypso version 16.1.
9.0	September 2019	Ninth edition – Updates for version 2.4.8.
10.0	September 2020	Tenth edition – Updates for version 2.6.13.
11.0	August 2021	Eleventh edition – Updates for version 3.11.0. Data Uploader is now an internal module.
12.0	October 2021	Edition 12 – Updates for version 3.12.0 – Added mapping for RateIndex_ISDA2021.
13.0	January 2022	Edition 13 – Updated for version 4.0.0, 4.0.1, 4.0.2, 4.0.3.
14.0	July 2022	Edition 14 – Updated for version 4.5.0.
15.0	January 2023	Edition 15 – Updated for version 4.5.0 – Added multiple Book and CP mapping
16.0	June 2023	Edition 16 – Updated for version 4.12.1



This document describes how to integrate Calypso with BloombergFIT Interface.

(1) NOTE: The Calypso License to use this Calypso Integration Module does not include a license for any third-party data services to which this module can interface. Clients are responsible for contracting with the appropriate third-party data service(s) prior to using this Calypso Integration Module.

I NOTE: The BloombergFIT interface is distributed as part of the Bloomberg-FIT module, which may support other Bloomberg interfaces that must be licensed separately.

IMPORTANT NOTE: For Cloud deployments please contact your application management team as the deployment procedure for Cap Cloud is different.



Table of Contents

Introduc	tion	6			
Setup		7			
2.1	Software Requirements	7			
2.2	Installation				
	 2.2.1 Data Uploader Installation 2.2.2 Calypso Components 2.2.3 QuickFIXJ Distribution 2.2.4 SSL Connection 2.2.5 TLS Connection 2.2.6 Proxy Connection 2.2.7 STunnel 2.2.8 Setup Config Data using Execute SQL 2.2.9 Message Workflow Setup 2.2.10 Task Station Setup 				
Data Ma	pping	13			
3.1	Legal Entity Mapping	13			
3.2	Parent Legal Entity Mapping	16			
3.3	Book Mapping	17			
3.4	Single Legal Entity Multiple Book Mapping	19			
3.5	Trader Mapping	19			
3.6	Rate Index Mapping	20			
Bloombe	erg FIX Engine Setup	21			
4.1	Configure the Engine	21			
4.2	Set Up the FIX Config File	21			
	 4.2.1 Sample Properties File 4.2.2 QuickFIXJ Settings 4.2.3 Calypso Settings 4.2.4 Out-of-Order FIX Message Settings 	21 22 22 22 23			
4.3	Set Up the gatewayservice Config File (Advanced)	23			
4.4	Launching the Bloomberg FIX Engine	23			
	 4.4.1 Adding Logging Categories				
Performa	ance and Stability Improvement	25			



5	5.1	Performar	Performance and Stability Feature					
5	5.2 Enabling Performance and Stability Feature							
Test 1	Tool	Setup: Fil	eWatcher2	7				
6	6.1	Setup the	FileWatcher Config File	27				
6	6.2	Launching	g FileWatcher	27				
		6.2.1 A 6.2.2 R	dding Logging Categories	27 27				
Test 1	Tool	Setup: Gl	2	8				
7	' .1	Set Up the	e GUI Config File2	8				
7	.2	Uploading	y Via the GUI2	8				
Keyw	ords		2	9				
Keyw Troub	ords blesh	ooting		9				
Keyw Troub 9	ords plesh 0.1	ooting Connectiv		9 81 81				
Keyw Troub 9 9	ords plesh 9.1 9.2	ooting Connectiv Message	2 vity	9 81 81				
Keyw Troub 9 9 9	ords blesh 0.1 0.2 0.3	ooting Connectiv Message Debug Lo	2 vity Processing gging	9 81 81 81				
Keyw Troub 9 9 9 9	ords blesh 0.1 0.2 0.3 0.4	ooting Connectiv Message Debug Lo Reporting	2 vity	9 81 81 81 81 81				
Keyw Troub 9 9 9 9 9 9	ords blesh 0.1 0.2 0.3 0.4 ndix.	ooting Connectiv Message Debug Lo Reporting	2 vity	9 81 81 81 81 81 81				
Keyw Troub 9 9 9 9 9 9 Apper	ords 0.1 0.2 0.3 0.4 ndix. 0.1	ooting Connectiv Message Debug Lo Reporting Engine Se	2 vity	9 81 81 81 81 81 81 83 83				



Introduction

The Bloomberg-FIT interface allows an End User Firm (Buy Side) to import trades which have been booked through the Bloomberg Terminal using Bloomberg's workflow.

The Calypso Bloomberg-FIT integration connects to Bloomberg using a FIX interface. Once the connectivity is setup, a trader can book a trade through the Bloomberg Terminal, and Bloomberg will send that as a FIX message to the Calypso Bloomberg interface. The message will then flow through the configured Calypso workflows which route the message through the appropriate stages to create a Calypso trade. Additional clearing lifecycle messages will also be sent over the FIX connection, and the appropriate lifecycle actions will be applied to the Calypso trade.

This document describes the configuration required to setup the workflows, etc. for the Bloomberg-FIT interface to run successfully.



Setup

2.1 Software Requirements

Please use the appropriate JRE version depending on the supported version for the base Calypso release you are running.

2.2 Installation

2.2.1 Data Uploader Installation

The use of the Bloomberg interface requires Data Uploader. All subsequent instructions assume that all Data Uploader installation steps have been completed successfully. This includes:

All Calypso versions:

- Applying the Gateway SchemaBase/SchemaData to your database
- Setting up the GATEWAYMSG and UPLOADESOURCEMSG workflows
- Setting up Task Station tabs for the workflows above
- Please refer to the Calypso Data Uploader Integration Guide for specific installation and configuration information. You **must** install and configure Calypso Data Uploader prior to configuring Bloomberg.

Calypso 16.1.0.84 and Bloomberg-FIT 3.11.0 and up:

• Data Uploader is an internal, optional module. It will automatically be installed when any dependent module, such as Bloomberg-FIT, is selected while installing.

2.2.2 Calypso Components

Simply choose the Bloomberg FIT component from within the Calypso Installer, or use the patch tool to add it to an existing installation.

Aside from installing the appropriate jars, the installation will also place files in the following directories:

- \$CALYPSO_HOME\client\bin\dbscripts
- \$CALYPSO_HOME\client\resources
- \$CALYPSO_HOME\docs\calypso-bloomberg-fit
- \$CALYPSO_HOME\legal

2.2.3 QuickFIXJ Distribution

The Bloomberg FIT interface uses the open source QuickFIXJ libraries for the FIX connectivity implementation. These libraries are included with the Calypso Data Uploader distribution and are installed automatically for you.



2.2.4 SSL Connection

The Bloomberg-FIT module supports secure communication over SSL, using Quick-FixJ library.

For SSL enabled FIX session, Bloomberg platform provides SSL certificates in either JKS or PEM format.

Adding 'SocketUseSSL=Y' setting to your FIX property file enables the SSL on FIX session.

If you need to use a specific SSL certificate, configure your session as shown below.

JavaKeyStore [JKS] Certificate

SocketUseSSL=Y

SocketKeyStore=/path/to/JKS/file/<your_key_identifier_name>.jks

SocketKeyStorePassword= [jks password]

PEM Certificate

Convert the PEM certificate to JKS certificate and use the generated JKS as mentioned above. For steps to convert PEM certificate to JKS, see the FAQ.

2.2.5 TLS Connection

The Bloomberg-FIT module supports secure communications over TLS using Quick-FixJ library

Bloomberg platform support FIX session on TLS - 1.2.

For TLS enabled fix session, Bloomberg platform provide TLS certificates in following format CMS, JKS, PEM and PKCS12 format.

BloombergFIT module is certified for TLS with PKCS12 format.

Adding 'SocketUseSSL=Y' setting to your FIX property file enables the TLS on FIX session.

If you need to use a specific SSL certificate, configure your session as shown below.

PKCS12 Certificate

SocketUseSSL=Y

SocketKeyStore=/path/to/JKS/file/<your_key_identifier_name>.pfx

SocketKeyStorePassword=[pkcs12 password]

KeyStoreType=PKCS12

EnabledProtocols=TLSv1.2

PEM Certificate

Convert the PEM certificate to PKCS12 certificate and use the generated PKCS12 as mentioned above. For steps to convert PEM certificate to PKCS12, see the FAQ.



2.2.6 **Proxy Connection**

The Bloomberg-FIT module supports proxy connection using Quick-FixJ library. That means FIX connection can be established via proxy server.

QuickFixJ-1.6.4 supports two types of proxies:

- SOCKS proxy
- HTTP proxy

SOCKS proxy can be used with and without SSL, whereas HTTP proxy only works without SSL.

Below is the quick-fix-j configuration for proxy:

```
# The [DEFAULT] section contains default settings for all sessions
# These are inherited by each session defined below unless they are overridden in the session settings
# If you're not sure about something, use the default for now
[DEFAULT]
ConnectionType=initiator
ReconnectInterval=10
HeartBtInt=20
LogonTimeout=20
LogoutTimeout=20
Calypso.LogOnInterval=5000
Calypso.LogOnRetryCount=5
# PROXY Support
ProxyType=socks
ProxyVersion=5
ProxyHost=<set proxy_host>
ProxyPort=<set_proxy_port>
```

Application software for HTTP and SOCKS proxy can be different and they are hosted in different IP and PORT. So, like HTTP, we have a separate configuration property for SOCKS in the CalypsoEnv property file.

HTTP Configurations	SOCKS Configurations
HTTP_PROXY_HOST	SOCKS_PROXY_HOST
HTTP_PROXY_PORT	SOCKS_PROXY_PORT
HTTP_PROXY_USERNAME	SOCKS_PROXY_USERNAME
HTTP_PROXY_PASSWORD	SOCKS_PROXY_PASSWORD

For proxy connectivity, our FIXEngine logic is:

- If property ProxyType in bloomberg-fit-fix.properties not empty
 - Check for proxy information present in FIX properties,
 - a. Use that to connect to Bloomberg via proxy.
 - Else check for proxy information in calypsouesr.properties
 - b. Use that to connect to Bloomberg via proxy.
 - Else throw an error.
- Else

ProxyUser=<set_proxy_user_password_if_any>
ProxyPassword=<set_proxy_user_password_if_any>



- Connect to Bloomberg directly.



2.2.7 STunnel

The Bloomberg-FIT module supports STunnel using Quick-FixJ library. That means FIX connection can be established via STunnel.

Assuming STunnel is installed, in trade-fix.properties, instead of pointing to Bloomberg platform, point to IP and port where STunnel is running. And in STunnel route this incoming to Bloomberg platform.

```
# In case of STunhel, this should point to IP where STunnel is running.
SocketConnectHost=127.0.0.1
# In case of STunnel, this should point to Port where STunnel is running.
SocketConnectPort=8822
# This should be set to your SenderCompID, as provided by Platform
SenderCompID=<SENDER_ID>
# This should be set to Tradeweb's TargetCompID, as provided by Platform
TargetCompID=<TARGET_ID>
```

The SSL certificates also can be added to the STunnel to encrypt the outgoing fix message.

For STunnel from stunnel.org, add the incoming source IP address in 'accept' and put IP of destination i.e. Bloomberg platform in 'connect' property. User can also specify the SSL/TLS certificates if required.

```
[bloomberg-proxy]
client = yes
accept = 127.0.0.1:8822 </P of source>
connect = 208.226.148.184:443 </P of destination (Bloomberg)>
cert = <Optional; SSL or TLS certificate path>
key = <Optional; SSL or TLS certificate password>
```

2.2.8 Setup Config Data using Execute SQL

When running Execute SQL, all necessary data files are uploaded.

- SchemaBase.xml
- GatewaySchemaBase.xml
- FIXSchemaData.xml
- FpMLSchemaData.xml
- BloombergFITSchemaData.xml

2.2.9 Message Workflow Setup

The Bloomberg module uses the UPLOADSOURCEMSG and GATEWAYMSG workflows when importing messages. These should have been set up as part of the Calypso Data Uploader Integration Guide.



Messages from the UPLOADSOURCEMSG workflow are translated from the external message format into Calypso's internal format and placed in the GATEWAYMSG workflow. The GATEWAYMSG workflow then translates the internal format, performs verifications, and saves the trade to the database.

Message Workflow Screenshots

The UPLOADSOURCEMSG workflow:



The GATEWAYMSG workflow:





2.2.10 Task Station Setup

The Bloomberg module uses the Data Uploader framework to create Task Station entries for all the messages and exceptions that are encountered. The user can view / reprocess the messages that are failed in validation from the task station.

Please refer to the Calypso Data Uploader Integration Guide for how to add the appropriate messages and exceptions to the Task Station.



Data Mapping

3.1 Legal Entity Mapping

The incoming Bloomberg FIX messages contain Legal Entity identifiers for all parties involved in the trade (Party, Counterparty, FCM, and CCP).

The Legal Entity for the Party, Counterparty, FCM, and CCP identifiers within the FIX message will be identified in Calypso using the Legal Entity Attribute 'BloombergFITParticipant', and the value would match the legal entity value provided in the FIX message.

The fetching logic will first search for a Legal Entity with the attribute value matching the value specified in the message, and if not found then it will search for a Legal Entity having a matching Short Name (case-sensitive or all uppercase).

🗾 Legal Entity- '	Version - 1 [143000/14300/	/calypso	_user]				
Utilities Help								
Short Name	BBGCLIENT				_	Status	Enabled	-
Full Name	BBG CLIENT					Role(s)	ProcessingOrg	
Parent						1		
Country	Country NONE							
🗾 Legal Entity A	ttributes Wi	ndow						<u>_ X</u>
Legal Enti	ty BBGCLI	ENT]	Role	ALL		•
Processing Org	ALL		-					
Attribute Typ	e ACCOL	NTING	-		Value			
Id Pro	cessing Org	Legal Entity	Role		Attribute	Туре	Attribute Valu	e
57202 ALL		BBGCLIENT	ALL	Bloomb	ergFITPa	rticipant	CALYPSO TECHNOLOGY	/, INC.

This lookup logic will be applied to PO, Counterparty, FCM, and CCP lookups. If no Calypso Legal Entity is found using the rules above, an error will be raised.

You can set a pipe () separated list in the LE attribute BloombergFITParticipant if you add Value = BloombergFIT to the domain "LEAttributeMultipleParticipant" so that multiple participants can be mapped to the same LE.

You can also map additional participants to the same legal entity using the following mapping:

Name = BloombergFIT/BloombergFITParticpant

Interface Value = <participant>



Calypso Value = <legal entity>

You can also map the participant by party, currency, currency pair, trader, source (incoming only, platform), product type (outgoing only).

The possible combinations need to be defined under Name = BloombergFIT/MappingConfig for Interface Value = Participant or OutgoingParticipant.

The possible values for Participant include:

Calypso Value = PARTY|CCY, PARTY|TRADER, SOURCE|PARTY, SOURCE|PARTY|CCY, SOURCE|TRADER|CCY, TRADER|CCY,PARTY,TRADER

The possible values for ParticipantOutgoing include:

Calypso Value = PARTY|TYPE|CCY, PARTY|CCY, PARTY|TYPE, PARTY, TYPE|TRADER, TRADER, TYPE|PARTY

Example:

Name = BloombergFIT/MappingConfig

Interface Value = Participant

Calypso Value = PARTY CCY

This means that the participant mapping values can be defined by party and currency.

Once this is defined, you can define the actual mapping values under: Name = BloombergFIT/BloombergParticipant or BloombergFIT/BloombergParticipantOutgoing Interface Value = <Any of the possible combinations defined under MappingConfig > Calypso Value = <Calypso counterparty>

Example: Name = BloombergFIT/BloombergFITParticipant Interface Value = PVNR|CAD Calypso Value = CPNYC This means that for Party = PVNR and Currency = CAD, the counterparty is CPNYC

Following is an example of mappings:



🔀 Calypso Mapping Window		
Calypso Mapping Window	Name: Interface Value: Calypso Value: Reverse Default: << Add >> Remove Configure Interfaces Configure Types	BloombergFIT/BloombergFITParticipantOutgoing
FiXBodyConstants FiXHeaderConstants FiXHeaderConstants FixTereContract MappingConfig Sook Book Sook Participant Participant		

23052209 JPY GlobalDEMO1

23052209|USD GlobalBFIT PVNR|BRL GlobalBFIT PVNR|CAD Global VCON|CALP GlobalDEMO2 BGD1|BRL CALYPSO_NYC BGD1|CAD CALYPSO

BGDM|JPY CALYPSO_LON

BGDMUSD CALYPSO_TOK

VCON CPY CPTY_1

- (I) [NOTE: For some products, the Legal Entity for the Party and Counterparty identifiers within the embedded FpML message and FIX message might need to be mapped correctly. User can map the FIX LegalEntity with the FpML LegalEntity using the Legal Entity Attribute 'BloombergFITFpMLParticipant', and the value would match the legal entity value provided in the embedded FpML message.
- You must contact Bloomberg to provide you with the internal identifiers contained in the messages. Alternatively, you may use the Bloomberg FIX engine to consume test trades booked in the Bloomberg TEST environment and use the details from the Task Station errors to determine the identifiers that must be set up as Calypso Legal Entity Attribute values.]



3.2 Parent Legal Entity Mapping

In some cases, there might be a need to map several separate Bloomberg identifiers to a single parent LE in Calypso.

To support this, first follow the procedure described above to setup separate Calypso Legal Entities each mapped to their corresponding Bloomberg identifier. Once that is complete, simply add the 'BloombergFITParent' attribute on each of the Calypso Legal Entities and populate it with the Calypso LE Short Name that you wish to be used as the parent.

When the lookup logic encounters a Legal Entity that has a parent defined, it will use the parent LE instead for all LE-related logic afterwards.

INOTE: This is only supported currently for the Party and Counterparty identifiers. In addition, when a parent is found, the child LE short name will be saved as a keyword on the trade; PlatformPO for the Party, PlatformCP for the Counterparty.]

🗾 Legal Entity- V	ersion - 1 [143	000/14300/calyp	so_user]			
Utilities Help						
Short Name	BBGDEALER_D4			- Stat	us Enabled	_
Full Name	BBG Dealer D4			- Role(s) CounterPart	ty
Parent						
Country	NONE		•			
🗾 Legal Entity At	tributes Windo	w				
Legal Entit	y BBGDEALER	L_D4		Role ALL		-
Processing Org	ALL		·			
Attribute Type		NG		Value		
Td F	rocessing Org	Legal Entity	Role	Attrib	ute Type	Attribute Value
57199 ALL	rocossing org	BBGDEALER_D4	ALL	BloombergFIT	Parent	BBGDEALER
57198 ALL		BBGDEALER_D4	ALL	BloombergFIT	Participant	D3
Legal Entity- V Utilities Help	ersion - 1 [143	000/14300/calyp:	so_user]			
Short Name	BBGDEALER			Stati	us Enabled	-
Full Name	BBG Dealer Mate	r		Role(s) CounterPart	ty
Parent						
Country	NONE		•			



3.3 Book Mapping

Once the Bloomberg interface determines the appropriate Legal Entity to use as the PO, it must then choose a Calypso Book for the trade. The section below outlines the logic used by the Bloomberg interface.

If no Calypso Book is found using the rules below, an error will be raised.

(1) [NOTE: When a Calypso Book is determined based on the lookup rules described below, the interface will verify that the Book's Legal Entity matches the Calypso Legal Entity found using the rules specified in the previous section. This is especially important to note when using a parent LE; the Book must belong to the parent LE, not the child.]

When a trade is booked in the Bloomberg Terminal the user is not able to set any Book or Account details. However, in Calypso we require the trade to be assigned to a Calypso Book for the PO.

To address this, the Bloomberg interface allows you to set a default book at the Legal Entity level, which will be used for all incoming Bloomberg messages where that Legal Entity is the PO. Specifically, the Calypso Book is determined using the Legal Entity Attribute 'BloombergFITBook' on the Legal Entity retrieved using the lookup rules described in the previous section.

The value in the 'BloombergFITBook' Legal Entity Attribute will be matched to a Calypso Book using the Book Attribute 'BloombergFITBook'. If no Book is found in Calypso with that attribute value, the fetching logic will look for a Book with the short name matching the 'BloombergFITBook' attribute value directly (case-sensitive).

	Legal Entity Attributes Window									
	Lega	al Entity	BBGCLIE	ENT			Role	ALL		T
	Processin	ng Org	ALL		•	·				
	Attribut	te Type	ACCOU	NTING	•	·	Value			
[Ъī	Proces	sina Ora	Legal Entity	Role	. [Attribute	Type	Attribute	Value
	57202	ALL	oing org	BBGCLIENT		Bloomb	eroFITP:	articipant	CALVESO TECHNO	LOGY INC
	59197			BBGCLIENT		Bloomb	ergEITBo	nok	BBGClieptDefault	codi, nici
	57177			DEGCETENT		Dioonio	cigi 110		bbacileneberadie	
	Rook Wind	low - Ver	sion -0 [14	3000/14300/ca	lynco i	iser]				
		IUW - YEI	51011-0 [14	3000/14300/08	iiypso_	1961]				
Vie	w Help									
		Book Id	57204		A	ttributes				
		Name	BBGBookDef	ault			Name		Value	
					A	cAdjustme	ntDays			
		Activity			A	cDateRule			*	
	0.000 m	ting Link	AEC		A	cReversalF	lule		T	
	ACCOUR				Ba	ankingBook			T	
	Leo	al Entity	BBGCLIENT		<mark>Bl</mark>	pombergFI	(Book		BBGClientDefault	
					Bo Bo	okBundle				



You set a comma-separated list in book attribute BloombergFITBook if you add Value = BloombergFIT to the domain "BookAttributeMultipleParticipant" so that multiple participants can be mapped to the same book.

You can also define mapping values by party, currency, currency pair, trader, source (incoming only, platform), product type (outgoing only).

The possible combinations need to be defined under Name = BloombergFIT/MappingConfig for Interface Value = Book or OutgoingBook.

The possible values for Book include:

Calypso Value = PARTY, PARTY|CCY, PARTY|TRADER, SOURCE|PARTY, SOURCE|PARTY|CCY, SOURCE|TRADER|CCY, TRADER, TRADER|CCY

The possible values for BookOutgoing include:

Calypso Value = PARTY TYPE CCY, PARTY CCY, PARTY TYPE, PARTY, TYPE TRADER, TRADER

Example:

Name = BloombergFIT/MappingConfig

Interface Value = Book

Calypso Value = PARTY CCY

This means that the book mapping values can be defined by party and currency.

Once this is defined, you can define the actual mapping values under: Name = BloombergFIT/BloombergFITBook or BloombergFIT/BloombergFITBookOutgoing Interface Value = <Any of the possible combinations defined under MappingConfig > Calypso Value = <Calypso book name>

Example: Name = BloombergFIT/BloombergFITBook Interface Value = PVNR|CAD Calypso Value = CADBOOOK This means that for Party = PVNR and Currency = CAD, the book is CADBOOK



3.4 Single Legal Entity Multiple Book Mapping

The Book Attribute 'BloombergFITBook' can also be used to map a single legal entity to several different books in Calypso using the incoming BIC codes. Set the relevant BIC code on each book as applicable. The books can have the same PO in Calypso.

[] [NOTE: This is only applicable for orders not created in Calypso.]

First it checks for the incoming BIC code to be mapped to a book and if found, the book is used and the Book LE is set as the PO. If no book is found with the BIC code, then it falls back to the logic of getting the PO from the BIC code.

🔀 Book Window -	/ersion -3 [161028/161/calypso_us	er]	-	- 🗆	×
View Help					
Book Id	26605	Attributes		Incoming	
Name	MyBank ABCD Book	Name /	Value	BIC code	
Activity	r	BB_TRADER_PORTFOLIO BankingBook	Ŧ	7	^
Accounting Link	TRADING	BloombergFITBook BookBundle	MBAB		
Legal Entity	PO	CAMoneyDiff Book CMF ID			
🔀 Book Window - V	/ersion -2 [161028/161/calypso_us	er]		- 0	×
View Help					
Book Id	26606	Attributes		Incoming	
Book Id	26606 MyBank WXYZ Book	Attributes	Value	Incoming BIC code	
Book Id Name Activity	26606 MyBank WXYZ Book	Attributes Name / BB_TRADER_PORTFOLIO BankingBook	Value	Incoming BIC code	^
Book Ic Name Activity Accounting Link	26606 MyBank WXYZ Book TRADING ~	Attributes Name / BB_TRADER_PORTFOLIO BankingBook BloombergFITBook PackExample	Value • MBWX	Incoming BIC code	

3.5 Trader Mapping

The Processing-Org trader present in fix message can be mapped to the TRADER available in Calypso. In the Calypso Mapping Window, the incoming trader name can be mapped to the existing TRADER in Calypso as shown below.



Zalypso Mapping Window		
E	Nama	PleasehaveEtT/Tvadave
	Nane;	
FIXBodyConstants	Interface Value:	C56650
FIXHeaderConstants	Calypso Value:	TRADER1
FutureContract MaturityMonthCode	Reverse Default:	
₽ PriceType	<< Add	
SecurityCode Traders	>> Remove	
🚽 C56650	Configure Interfaces	1

But if no mapping is found, then By Trader is set with what is present in the incoming message.

[] [NOTE: This Calypso mapping will be used for outgoing FIX messages.]

3.6 Rate Index Mapping

Mapping category RateIndex_ISDA2021 to handle the index mappings from ISDA2021.

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

Sample mapping:

Name = BloombergFIT/RateIndex_ISDA2021

Interface Value = USD-SOFR-OIS Compound

Calypso Value = USD~SOFR~COMPOUND

If not defined, the FpML mapping is used instead, if defined.

Name:	FpML/RateIndex_ISDA2021				
Interface Value:	USD-LIBOR				
Calypso Value:	USD~LIBOR~T3750				
Reverse Default:					



Bloomberg FIX Engine Setup

The Bloomberg FIX Engine is responsible for getting messages from the Bloomberg platform and handing it off to the appropriate workflows. It can also be run in 'Test' mode to only read FIX messages from files, which can be useful for replaying problematic messages or for Unit Testing. (Please see Test Tool Setup: FileWatcher for more details.)

The Bloomberg FIX Engine is built on the Calypso FIX Engine framework, and therefore while setting it up you will find generic FIX setup vs. Bloomberg specific setup. For clarity, all required steps are listed below.

Please review the standard Calypso documentation for Engine setup to read about several useful engine parameters (such as thread count) and how to set them.

4.1 Configure the Engine

All the database-based engine configuration is completed as part of applying the schema, including the engine name, event subscription, event filter, and event policy, as well as assigning a unique id to the Engine.

You may refer to the schema file for more details. In addition, the Appendix contains screenshots of the engine configuration screens.

4.2 Set Up the FIX Config File

To run the Bloomberg FIX Engine out-of-the-box you will need a properties file with the name "bloombergfitfix.properties" with the appropriate FIX connection settings.

A sample file is included under \$CALYPSO_HOME/client/resources with the name "bloombergfit-fix.properties.sample". You will need to rename the file to "bloombergfit-fix.properties".

Note that, as previously mentioned, the Bloomberg FIX Engine uses the QuickFIXJ library for the FIX connectivity implementation. The QuickFIXJ library has many options that can be configured on a FIX session, using a standard properties file. The Calypso Bloomberg FIX Engine uses this same file for internal settings as well.

For simplicity, we have provided a sample bloombergfit-fix.properties file and will only refer to the minimum settings that must be changed to work with Bloomberg connectivity. You can view all the available settings on the QuickFIXJ Configuration page located on their documentation site at:

http://www.quickfixj.org/quickfixj/usermanual/1.5.3/usage/configuration.html

4.2.1 Sample Properties File

The sample "bloombergfit-fix.properties" file appears similar to the following example:

```
# Default settings for sessions.
# These are inherited by each session defined below
# unless they are overridden in the session settings.
[DEFAULT]
ConnectionType=initiator
```



ReconnectInterval=10

```
HeartBtInt=20
LogonTimeout=20
Calypso.TestMode=false
# DUMMY_SEFCLIENT session definition (Bloomberg SEF client)
[SESSION]
Calypso.FIXMessageType=BloombergFIT
BeginString=FIX.4.4
SenderCompID=DUMMY_SEFCLIENT
TargetCompID=BLPC
DataDictionary=DD_BloombergFIT.xml
SocketConnectHost=205.216.112.15
SocketConnectPort=20036
FileLogHeartbeats=Y
FileIncludeMilliseconds=Y
```

```
FileIncludeTimeStampForMessages=Y
```

```
StartTime=07:00:00
EndTime=23:00:00
TimeZone=America/New York
```

4.2.2 QuickFIXJ Settings

To connect with Bloomberg successfully, you will need to change the SenderCompID, TargetCompID, SocketConnectHost, and SocketConnectPort connection properties to the correct values for your setup. Please contact Bloomberg support for these details.

Additional points to note regarding the core QuickFIXJ settings:

- The QuickFIXJ settings allow you to configure multiple sessions in a single properties file. This means if you have multiple Bloomberg session logins, you can use a single Bloomberg FIX Engine to connect to all of them.
- FileStorePath and FileLogPath are defaulted to \$USER_HOME/Calypso/FIXEngine/Store and \$USER_HOME/Calypso/FIXEngine/Log respectively. These may be overridden at the DEFAULT or SESSION level within the config file. There is no support for other Store or Log mechanisms at this time.

4.2.3 Calypso Settings

There are two Calypso settings to note from the sample:

• Calypso.TestMode: This setting should be placed in the DEFAULT section of the properties file. If set to 'true', the Bloomberg FIX Engine will **not** connect to ANY session defined within the properties file. This can be used when using the File Watcher component to load the FIX messages via files rather than connecting to Bloomberg. If not set to 'true', the engine will support **both** messages from Bloomberg as well as from files.



• Calypso.FIXMessageType: This setting should be placed in the SESSION section(s) of the properties file. The value **must** be 'BloombergFIT' for Bloomberg-SEF sessions.

4.2.4 Out-of-Order FIX Message Settings

In order to process FIX messages that are not in the correct sequence, you need to set the following properties to disable the validation:

- ValidateUnorderedGroupFields=N
- ValidateFieldsOutOfOrder=N

4.3 Set Up the gatewayservice Config File (Advanced)

In addition to viewing exception messages in the Task Station, you may also configure Bloomberg to write all Success or Reject messages to the file system. This is configured within the gatewayservice.properties file distributed with Data Uploader.

A sample file is included under \$CALYPSO_HOME/client/resources with the name ".sample" as the suffix. To configure messages to be written to the file system, set the following in the config file:

```
BloombergFITPublishers=File,RejectedFile
```

At the minimum, you must have an empty entry (i.e. BloombergFITPublishers=) to avoid any spurious errors in the log.

You can customize the FilePublishToLocation property to control where these files are written.

4.4 Launching the Bloomberg FIX Engine

4.4.1 Adding Logging Categories

To see logging messages for the Data Uploader and Bloomberg modules you need to set the following log categories:

- UPLOADER: Set this to see logging for the Data Uploader translation from the internal Calypso xml format to the actual trade object.
- BloombergFIT: Set this to see logging for the Bloomberg translation from the external format to the internal Calypso xml format.
- FIX: Set this to see logging for the shared FIX connectivity & message processing pieces of the Bloomberg FIX Engine.

Additional debugging categories are listed in Troubleshooting.

4.4.2 Running the Bloomberg FIX Engine

With the previous steps completed, you are now ready to run the Bloomberg FIX Engine.

To start/stop the Engine server, use the DevOps Center or the Engine Manager in Web Admin.



With the Bloomberg Engine operating, you can then allege/affirm trades through the Bloomberg Terminal. The Bloomberg Engine will process the trade messages and create corresponding trades in Calypso.

The Task Station will display any errors that may occur.

4.4.3 Daily Stop / Restart

The Bloomberg FIX server's shutdown daily after business hours and startup again at the start of business the next day. As part of this daily cycle, the Sequence Numbers for the FIX connections are reset as well.

The Calypso Bloomberg FIX Engine handles this for you automatically, based on the values set in the fix.properties settings file for the properties StartTime, EndTime, and TimeZone. These properties control when the engine determines that a new session should be started, and the Sequence Numbers reset.

> For more details on these settings, please refer to the QuickFIXJ documentation site.



Performance and Stability Improvement

5.1 Performance and Stability Feature

To improve the performance for message processing we have now the provision to perform the upload process in the Engine side via the API instead of Data Server via the workflow rules. Workflow rules are executed in the Data Server, thus keeping Data Server busy every time data is uploaded. Client side execution has the advantage of using the API and cache. And it can thus perform better and Data Server is free for performing other tasks.

The new enhancement provides the equivalent features that were available via the workflow based approach such as:

- Persistence of external messages as BO messages
- Ability to Re-process failed messages
- Maintain the order in which the messages are received.
- Acknowledgement generation

To enable the new approach, we have introduced two new properties in "fix.properties" and "bloombergfituploader.properties". Please note that if these are not set, we will default to the current way of processing in Data Server via the workflow rules which is the BOMessage mode.

The following are the properties:

- uploadMode: Possible values are 'BOMessage' and 'Local', default value is 'BOMessage'.
 - BOMessage: It is self-explanatory; it is the workflow-based model that every interface currently uses.
 - Local: It is for using the API, however, when using the Local mode, we need another property "persistMessages".
- persistMessages: Applicable only when "uploadMode" property is set to 'Local'. Possible values are 'All', 'None' and 'Failure', default value is 'None'.
 - All: External messages are always persisted as BO messages.
 - None: External messages are _NOT_ persisted as BO messages. If the message fails in translation or validation, the message needs to be resent or handled via custom code.
 - Failure: External Messages are persisted only in case of failure in translation or validation. This is the _recommended_ configuration for the 'Local' mode as this will not save any BO messages in case it is all processed fine and only create messages in case of any translation failures which will enables the failed messages to be reprocessed as before and also improve performance by not saving the success BO messages.



#Valid combination for uploadMode and persistMessage

π					
#					
#				PERSIST MESSAGES	
#					
#			None	Failure	All
#					
#	M	Local	No BOMessage	BOMessage will be created	BOMessage will be
#	0		will be created	only in case of failure	always created
#	D				
#	E	BOMessage	Not Applicable	Not Applicable	BOMessage will be
#	S				always created
#					
#					

For better performance it is recommended to use uploadMode as Local with persistMessages as Failure.

Please note that in the 'Local' mode for message reprocessing "**UpdateManagerEngine**" needs to be running. Failed messages will be stuck in "PENDING" status and pending Messages are re-processed via the UpdateManagerEngine. The message workflows are changed to generate an event 'PSEventUploadReprocess' every time a failed message is re-processed. Update Manager Engine would then receive these events and processes them again and generate acknowledgement if needed.

The following Events are required by the Update Manager Engine.

- PSEventUploadReprocess
- Refer to the Appendix for UpdateManagerEngine config.

5.2 Enabling Performance and Stability Feature

- » Run the schema data with (GatewaySchemaBase.xml and GatewaySchemaData.xml).
- » Import the workflows (GATEWAYMSG.wf, UPLOADSOURCEMSG.wf).
- » Change the properties in "fix.properties" and "bloombergfituploader.properties file as explained above.
- » Configure the Update Manager Engine.



Test Tool Setup: FileWatcher

() [NOTE: The details in this section are provided for testing purposes only, and not recommended for production use.]

As mentioned in the previous section, the Bloomberg FIX Engine also supports processing FIX messages from files. To achieve this, you must run the DataUploader FileWatcher in 'bloombergfit' mode, so that it will load files from a specified location and pass them on to the Bloomberg FIX Engine.

The steps below assume you've already set up the Data Uploader module as per the Calypso Data Uploader Integration Guide, (in Calypso 16.1.0.84 with Bloomberg-FIT 3.11.0 and higher, Data Uploader is an internal module.)

6.1 Setup the FileWatcher Config File

To run the FileWatcher in 'bloombergfit' mode, you will need a properties file with the appropriate settings. A sample file "bloombergfituploader.properties" is included under \$CALYPSO_HOME/client/resources with the name ".sample" as the suffix.

Please change the fileDir as required for the polling directory. Also note that the fixSettings property must point to the Bloomberg FIX Engine's property file.

For other details, please refer to the Calypso Data Uploader Integration Guide.

6.2 Launching FileWatcher

6.2.1 Adding Logging Categories

To see logging messages for the Data Uploader and Bloomberg modules you need to set the following log categories:

- UPLOADER: Set this to see logging for the Data Uploader FileWatcher component.
- BloombergFIT: Set this to see logging for the Bloomberg FileWatcher component.
- FIX: Set this to see logging for the shared FIX FileWatcher component.

6.2.2 Running FileWatcher

With the previous steps completed, you are now ready to run the FileWatcher.

To start/stop the Engine server, use the DevOps Center or the Engine Manger in Web Admin.

With the FileWatcher operating, you can then place a '.fix' file to import in the watched directory (specified by fileDir in the properties file). At the end of the current wait interval, the FileWatcher will notify the Data Uploader which will then load the file and hand it off to the Bloomberg FIX Engine.

The Task Station will display any errors that may occur.



Test Tool Setup: GUI

() [NOTE: The details in this section are provided for testing purposes only, and not recommended for production use.]

The Calypso Bloomberg-FIT Interface is built on the Data Uploader framework, and therefore supports uploading Bloomberg FIX files through the Data Uploader GUI. This can be useful for testing, and does not require you to run the Bloomberg FIX Engine.

The steps below assume you've already setup the DataUploader module as per the Data Uploader Setup Guide, including adding the GUI window to your menu.

7.1 Set Up the GUI Config File

[NOTE: The need for this step will be removed in a future release.]

To upload Bloomberg FIX files through the GUI, you will need a properties file with the appropriate settings. A sample file "bloombergfit-datauploader-gui.properties" is included under \$CALYPSO_HOME/client/resources with the name ".sample" as the suffix.

Note that the fixSettings property must point to the Bloomberg FIX Engine's property file, although the engine itself does not need to be running.

7.2 Uploading Via the GUI

With the previous steps completed, you are now ready to upload Bloomberg FIX files using the GUI. Simply launch the Data Uploader GUI from the menu, choose the Source/Format, browse to select your '.fix' file, and upload.

For further details on using the Data Uploader GUI, please refer to the Calypso Data Uploader Integration Guide.

[NOTE: The uploaded file must have a '.fix' extension, not '.xml'.]





Keywords

In addition to the keywords listed above in the "Trade Workflow" section, the Calypso standard Clearing keywords, as well as Bloomberg specific keywords, will be populated for incoming trades accordingly. A full list is provided in the table below.

() [NOTE: This is an early release version and keyword names are subject to change. Such changes will be documented in the Release Notes, so please ensure you review them whenever upgrading from previous Bloomberg interface versions.]

Keyword Name	Description	Comments
TradeSource	Always set to 'BloombergFIT'	Used for engine logic behind the scenes; do not change.
PlatformAPIUser	FIX Session Id	Used for engine logic behind the scenes; do not change.
USIPrefix	Bloomberg Regulatory Reporting value	
USIValue	Bloomberg Regulatory Reporting value	
ССР	LE short name of CCP	
CCPClearingBroker	The clearing broker (when available in the trade)	
OriginalCounterparty	Bilateral counterparty	Set on the cleared trade, to show the original counterparty before novation.
Platform	Always set to 'BloombergFIT'	
PlatformStatus	Bloomberg Trade Status	
PlatformTransactionId	Bloomberg Transaction Id	Set to SecondaryOrderID which connects all messages for a single trade.
PlatformTradeld	Bloomberg Trade Id	As per Bloomberg internal identifier.
PlatformOrderld	Bloomberg Order Id	
CCPStatus	Sending, Cleared, Rejected	
CCPClearedDate	Date when trade was cleared by CCP	



Keyword Name	Description	Comments
CCPMessageTimestamp	GMT timestamp of last message to/from CCP	
CCPTradeID	CCP assigned deal id	
PriorUSIPrefix	Bloomberg Regulatory Reporting value	Set on a cleared trade, with the USI Prefix of the original bilateral trade.
PriorUSIValue	Bloomberg Regulatory Reporting value	Set on a cleared trade, with the USI Value of the original bilateral trade.
PlatformRejectReason	Reject reason provided in Bloomberg message	
PlatformPO	The PO Legal Entity short name which maps to the original Bloomberg message value	This is only set if the Calypso LE was pointing to a Parent LE, and therefore the child LE is preserved as a keyword.
PlatformCP	The CP Legal Entity short name which maps to the original Bloomberg message value	This is only set if the Calypso LE was pointing to a Parent LE, and therefore the child LE is preserved as a keyword.
AllocationClearingType	Specifies whether this trade is part of a Pre-Allocation or Post-Allocation allocation workflow	Set to "Pre-Allocation" if pre-allocation, "Bunched-Order" if bunched-order, and empty if this is not part of an allocation workflow at all.
BlockUSIPrefix	Bloomberg RegReporting value	Set on an allocation leg trade, with the USI Prefix of the allocation block trade.
BlockUSIValue	Bloomberg RegReporting value	Set on an allocation leg trade, with the USI Value of the allocation block trade.
CCPConfirmationId	Bloomberg CCP confirmation Id	
ClearingOrderId	Bloomberg clearing order Id	



Troubleshooting

This section contains details on how to troubleshoot any issues you may encounter with the Bloomberg Engine or processing messages, and what to do if you need to report an issue to Calypso.

9.1 Connectivity

The Bloomberg FIX Engine automatically attempts to reconnect if a connection with the Bloomberg platform is lost. On reconnect, it will first process any queued messages, and then be available to process new messages.

In case messages are not being received by the engine, please check the log files produced by the QuickFIXJ library. As mentioned previously, the default location for these logs is \$USER_HOME/Calypso/FIXEngine/Log. You can also check the Calypso Bloomberg FIX Engine logs.

9.2 Message Processing

For message processing failures, check the Task Station of UPLOADSOURCEMSG or GATEWAYMSG workflow errors. Please review the installation section of this document for details.

In addition, if you have configured the gatewayservice.properties file to log message processing errors to files, you can review those files as well.

9.3 Debug Logging

Additional logging can be configured to help with debugging errors; just set the following log categories:

- FIX_DEBUG_XML: Set this category to create xml files of the incoming messages from the Bloomberg platform as well as the interim Calypso xml format created by the translation under \$USER_HOME/Calypso/Bloomberg. Note that these files can be used for loading via the File Watcher.
- FIX_DEBUG_API: Set this category to see additional logging statements for the FIX connectivity, including Login/Logout and Admin messages (such as resend requests). These logging statements are helpful to debug any FIX connectivity issues, as well as see headers of the messages coming in from the Bloomberg platform before they are interpreted by the Bloomberg FIX Engine.

9.4 Reporting Issues to Calypso

If, after attempting all the above steps you still need to contact Calypso, please ensure that your ticket contains the following information:

- All available logs, including:
 - QuickFIXJ connectivity logs
 - Bloomberg FIX Engine logs
 - Bloomberg FileWatcher logs



- DataServer and EventServer logs
- FIX messages, if applicable
- Clear description of the issue, including:
 - What steps were executed to produce the issue, both in the Bloomberg Terminal and Calypso
 - What attempts, if any, were made to debug the issue, and what were the results

Appendix

10.1 Engine Setup

DataUploader Engine for uploading Bloomberg-FIT files

Image: Configuration: Persisted Event Configuration: Provide State of Start on Starture: Start on Starture: <th>Server Administration ×</th> <th></th> <th></th> <th></th> <th></th> <th>Shams -</th> <th>- 0</th>	Server Administration ×					Shams -	- 0
Server Metrics	- → C 🗋 localhost:8180)/engineserver/adr	nin/manage/editEngine	e?engineName	e=Bloor	nbergFITUploa 🖞	2
Nerver Netrics Profiler Manage Log Engine Class: Engine Class: Engine DataUploaderEngine Display Name: Engine DataUploaderEngine Display Name: Elioomberg-FIT Uploader Persisted Event Configuration: PSEventAccountBilling Persisted Event Configuration: PSEvent Filter: AltTransfersKnownEventFilter Engine Start on Startup: Engine Startup: Engin						1	
Environment: 14100 User: calypso_user [Lgo Engine Name: Engine ID: Max Queue Size: Max Batch Size: Number of Threads: Image: I	Server+ Metrics+	Profiler v Man	age v Logs				PSO
Engine Configuration BiombergFITUploader Ingine Name: Engine Class: com.calypso.tk.engine.DataUploaderEngine Xsplay Name: Type: BiombergFITUploader Type: BiombergFITUploader Pricing Environment: Pricing Environment: Save settle position changes: BiombergFITUploader Parsisted Event Configuration: PSEventAccountBilling Verk Filter: AllTransfersKnownEventFilter Ingine Manager Configuration: Start on Startup: enginesenver Ingine Manager Configuration: Start on Startup: Ingine Manager Configuration: St				Environment	t: 14100	User: calypso_user	[Logou
Engine Configuration: Ingine Anne: Engine ID: IsombergFITUploader T21 ingine Class: Event Pool Policy: Pricing Environment: commongFITUploader Engine Engine Berver Stopper FIT Uploader Type: Bloomberg-FIT Uploader Engine Berver Persisted Event Configuration: Procention: PSEventAccountBilling Imagine Annager Configuration: Vevent Filter: AllTransfersKnownEventFilter AllTransfersKnownEventFilter Imagine Start on Startup: engineserver Start on Startup: engineserver Start on Startup: Preverse EngineSertup Configuration. Vere Lige Monery Provide Start Start on Startup: engineserver Imagine Annager Configuration: Prevere Start Imagine Annager Configuration: Prevere Start Imagine Annager Configuration: Start on Startup: Imagine Annager Configuration: engineserver Imagine Annager Configuration: engineserver Imagine Annager Configuration: engineserver Imagine Annager Configuration: engineserver Imagine Annager Configu							
Engine Name: Engine ID: BloombergFITUploader [121 ingine Class: Pricing Environment: com.calypso.tk.engine.DataUploaderEngine Type: BloombergFITUploader Type: Bloomberg-FIT Uploader EngineServer Description: EngineServer Bloomberg-FIT Uploader Pricing Environment: Prested Event Configuration: Pricing Environment: PSEventAccountBilling Image:	Engine Configuration						
BloombergFITUploader BloombergFITUploader 121 Engine Class: Pricing Environment: com.calypso.tk.engine.DataUploaderEngine Type: Steplay Name: Type: BloombergFITUploader Engine EngineServer Description: EngineServer Bloomberg-FIT Uploader Configuration attributes PROJECTED_DAYS ReversAL_CRE Start UP TWE OUT_RESTART USE_BOOK_PRICING_ENV VALUATION_TIMES VALUATION_TIMES VALUATION_TIMES VALUATION_TIMES VALUATION_CHECK XFER_NEXT_EVER_BV XFER_NEXT_EVENT XFER_NEXT_EVENT XFER_NEXT_EVENT XFER_NEXT_EVENT XFER_USE_ADAMEY DIFF XFER_USE_POS_AGGREGATION_ONLY XFER_USE_	Engine Name: 😮	Engine ID:	Max Oueue Size: 😗	Max Batch :	Size: 😮	Number of Thre	ads: 😗
ingine Class: toom.calypso.tk.engine.DataUploaderEngine Xisplay Name: Type: BloombergFITUploader Engine Description: Bloomberg-FIT Uploader ProjecteD_DAYS ReveRsAL_CRE Start DP PSEventAccountBilling Vent Filter: AllTransfersKnownEventFilter ingine Manager Configuration: enginesenver ▼ Configuration: Start on Startup: AllTransfersKnownEventFilter PSEventAccountBilling Configuration: Start on Startup: AllTransfersKnownEventFilter PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountFilter PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling Configuration: PSEventAccountBilling PSEventAccountBillin	BloombergFITUploader	121					
com.calypso.tk.engine.DataUploaderEngine FKEngine default Xisplay Name: Type: Save settle position changes: Save settle position changes: Save settle position changes: Image: Configuration attributes ProjecteD/DAYS ProjecteD/DAYS Persisted Event Configuration: FK/Engine Image: Configuration attributes PSEventAccountBilling Image: Configuration attributes Start UP Vent Filter: Image: Configuration: Start on Startup: Valuation_TimeZones VERSION_CHECK Valuation_TimeZones Valuation: Start on Startup: VFER_CHECK_FIRST VFER_VER_GEV VFER_USE_AUTOMATIC_ACCOUNT VFER_USE_AUTOMATIC_ACCOUNT VFER_USE_AUTOMATIC_ACCOUNT VFER_USE_MONEYOIFF VFER_USE_MONEYOIFF VFER_USE_BEVERSE Verence Image: Bloombergfituploader.propertik VFER_USE_REVERSE	ngine Class:	1	Event Pool Policy: 😗	1 [Pricing En	vironment: 😗	
Absplay Name: Type: BloombergFITUploader Engine EngineServer Description: Image: Configuration: Persisted Event Configuration: Image: Configuration: PSEventAccountBilling Image: Configuration: Vent Filter: Image: Configuration: AllTransfersKnownEventFilter Image: Configuration: Image: Manager Configuration: Start on Startup: engine Manage	com.calypso.tk.engine.DataUpload	derEngine	FIXEngine	•	default	•	
BloombergFITUploader Engine EngineServer Description: Configuration attributes ProJecTED_DATS ReVERSAL_CRE STARTUP TWEOUT_RESTART USE_BOOK_PRICING_ENV VALUATION_TIMES VALUATION_TIMES VALUATION_TIMES VALUATION_TIMES VALUATION_CHECK XFER_CHECK_FIRST XFER_NEVER_BV XFER_NEVER_BV XFER_NEVER_BV XFER_NEVER_BV XFER_NEVER_BV XFER_NEVER_BV XFER_NEVER_BV XFER_NEVER_BV XFER_NEVER_T XFER_NEVER_BV XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_MONEYDIFF XFER_USE_NONEYDIFF XFER_USE_REVERSE config bloombergfituploader.propertik ▼)isplay Name: 😮	Type:	Save settle position o	:hanges: 😮			
Siloomberg-FIT Uploader ersisted Event Configuration: PSEventAccountBilling PSEventAccountBilling Imagine Manager Configuration: Start on Startup: engineserver ▼ Configuration: Start on Startup: PSEventAccountFilter Configuration: Start on Startup: engineserver ▼ Configuration: Start on Startup: engineserver ▼ Configuration: Start on Startup: engineserver ▼ Configuration: Start on Startup: engineserver ▼ Configuration: Start on Startup: engineserver ▼ Configuration: Start on Startup: engineserver ▼ Configuration: Start on Startup: Configuration: Configuration: Start on Startup: Configuration: <td>BloombergFITUploader Engine</td> <td>EngineServe</td> <td>r 🔍</td> <td></td> <td></td> <td></td> <td></td>	BloombergFITUploader Engine	EngineServe	r 🔍				
PROJECTED_DAYS PROJECTED_DAYS PROJECTED_DAYS PROJECTED_DAYS PROJECTED_DAYS PROJECTED_DAYS REVERSAL_CRE STARTUP TIMEOUT_RESTART USE_BOOK_PRICING_ENV VALUATION_TIMES VALUATION_TIMEZONES V	escription:		Configuration attribu	tes			
REVERSAL_CRE start UP TIMEOUT_RESTART USE_BOOK_PRICING_ENV VALUATION_TIMES VALUATION_T	loomberg-FIT Uploader		PROJECTED_DAYS				*
Persisted Event Configuration: STARTUP PSEventAccountBilling Image: Start on Startup: went Filter: Image: Configuration: AllTransfersKnownEventFilter Image: Configuration: ingine Manager Configuration: Start on Startup: engineserver XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE Config bloombergfituploader.propertie			REVERSAL_CRE				
PSEventAccountBilling	Persisted Event Configuration:		STARTUP				
USE_BOOK_PRICING_ENV VALUATION_TIMES VALUATION_TIMES VALUATION_TIMEZONES VALUATION_TIMEZONES VALUATION_TIMEZONES VERSION_CHECK XFER_CHECK_FIRST XFER_NEXT_EVENT XFER_NEXT_EVENT XFER_NEXT_EVENT XFER_NEXT_GENERATION XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_MONEYDIFF XFER_USE_POS_AGGREGATION_ONLY XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE config bloombergfituploader.propertik	PSEventAccountBilling	• •	TIMEOUT_RESTART				
VALUATION_TIMES vent Filter: AIITransfersKnownEventFilter VALUATION_TIMEZONES VALUATION_TIMEZONES VALUATION_TIMEZONES VERSION_CHECK XFER_CHECK_FIRST XFER_NEVER_BV XFER_NEXT_EVENT XFER_NEXT_EVENT XFER_PAST_GENERATION XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_MONEYDIFF XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE config bloombergfituploader.propertit			USE_BOOK_PRICING_	ENV			
VALUATION_TIMEZONES VALUATION_TIMEZONES VERSION_CHECK XFER_OHECK_FIRST XFER_NEXER_BV XFER_NEXT_EVENT ATFER_NEXT_EVENT XFER_NEXT_GENERATION XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_MONEYDIFF XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE config bloombergfituploader.propertik			VALUATION_TIMES				
vers Filter: AllTransfersKnownEventFilter VersioN_CHECK VersioN_CHEC			VALUATION TIMEZO	NES			
AllTransfersKnownEventFilter	vent Filter:		VERSION CHECK				
AFER_NEVER_BV XFER_NEVT_EVENT angine Manager Configuration: Start on Startup: Argine Manager Configuration: Startup:	AllTransfersKnownEventFilter					1.1	
AFEC_NEVEC_DV XFER_NEXT_EVENT angine Manager Configuration: Start on Startup: AFER_USE_AUTOMATIC_ACCOUNT XFER_USE_MONEYDIFF XFER_USE_POS_AGGREGATION_ONLY XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE config bloombergfituploader.propertit							
AFER_NEXT_EVENT ngine Manager Configuration: Start on Startup: angineserver ▼ Start on Startup: XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_MONEYDIFF XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE config bloombergfituploader.properti€ ▼			AFER_NEVER_DV				
ngine Manager Configuration: Start on Startup: ArER_PAST_GENERATION ×FER_USE_AUTOMATIC_ACCOUNT ×FER_USE_MONEYDIFF ×FER_USE_POS_AGGREGATION_ONLY ×FER_USE_REVERSE config bloombergfituploader.properti€ ▼			T AFER_NEXT_EVENT				
engineserver XFER_USE_AUTOMATIC_ACCOUNT XFER_USE_MONEYDIFF XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE bloombergfituploader.propertie config bloombergfituploader.propertie	ngine Manager Configuration:	Start on Start	up: XFER_PAST_GENERAT	ION			
XFER_USE_MONEYDIFF XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE config bloombergfituploader.propertite	engineserver 🔻		XFER_USE_AUTOMAT	IC_ACCOUNT			
XFER_USE_POS_AGGREGATION_ONLY XFER_USE_REVERSE config bloombergfituploader.propertite Delete Engine			XFER_USE_MONEYDIF	F			
XFER_USE_REVERSE config bloombergfituploader.propertite			XFER_USE_POS_AGGF	EGATION_ONLY			
config bloombergfituploader.propertie			XFER_USE_REVERSE				
Delete Engine Save			config		bloombe	rgfituploader.properti	ε 🖵
	Delete Engine					Cancel	Save



FIX Engine for uploading messages from Bloomberg platform.

Engine Configuration

Engine Name: 😮	Engine ID:	Max Queue Size: 😗	Max Batch Size: 🤇	Number of T	'hreads: 😗
FIXEngine	116				
Engine Class:		Event Pool Policy: 😗	Pricing	g Environment: 😗	
com.calypso.tk.engine.FIXEngine		FIXEngine	▼ INTR/	ADAY 🔻	
Display Name: 🥹	Application Type:	Save settle position chang	es: 🕜		
FIX Engine	EngineServer	•			
Description:		Configuration attributes			
		REVERSAL_CRE			^
Persisted Event Configuration:		STARTUP			
PSEventAccountBilling	• O O	TIMEOUT_RESTART			
PSEventFIXMessage	-	USE_BOOK_PRICING_ENV			
		VALUATION_TIMES			
	Ψ.	VALUATION_TIMEZONES			
Event Filters:	- 00	VERSION_CHECK			
FIXEngineEventFilter		XFER_CHECK_FIRST			
· · · · · · · · · · · · · · · · · · ·		XFER_NEVER_BV			_
	_	XFER_NEXT_EVENT			
Engine Manager Configuration:	Start on Startup:	XFER_PAST_GENERATION			
engineserver T		XFER_USE_AUTOMATIC_AC	COUNT		
		XFER_USE_MONEYDIFF			
		XFER_USE_POS_AGGREGAT	ION_ONLY		
		XFER_USE_REVERSE			
		config	fix.pr	operties	-
Delete Engine				Cancel	Save



Update Manager Engine

Server +	Metrics •	Profiler v	Manage v	Monitoring ,	Logs	4	CALYI	PSO®
						Environment: TRUNK	User: calypso_user	[Logout]

Engine Configuration

Engine Name: 😗	Engine ID:	Max Queue Size: 😗 🛛 🗚	Wax Batch	Size: 😗	Number of Threads: 😗	
UpdateManagerEngine	107					
Engine Class:		Event Pool Policy: 🛛 Pricing Environment: 🕄				
com.calypso.tk.engine.UpdateManagerEr	ngine		•	default	v	
Display Name: 😮	Application Type:	Save settle position change	is: 😮			
Update Manager Engine	EngineServer	•				
Description:		Configuration attributes				
		Attribute Name		Attribute	Value 🔶	
Persisted Event Configuration:		BALANCE_MODE				
PSEventAccountBilling	• O O	CLASS_NAME				
PSEventRepublish	*	DISPLAY_NAME				
PSEventUploadReprocess		DateType				
	-	EVENT_ORDER				
Event Filters:		EXCLUDE_PRODUCTTYPE				
AllTransfersKnownEventFilter	- C O	EXCLUDE_STATUS				
UpdateManagerEngineEventFilter	^	HANDLE_FUTURE_LIQ_CASH	L_FLOWS			
		IGNORE_ACTION				
	T	INSTANCE_NAME				
Engine Manager Configuration:	Start on Startup:	INV_MAX_POSITION				
engineseiver		LIQUIDATION_TIMEOUT				
		MAX_TIMER_POSITION				
		MCC_DATE_KEYWORD				
		MCC_FEED_NAME			-	
Delete Engine				I	Cancel Save	



FAQ

- 1. Unable to connect to Bloomberg platform.
- Ans: It can be a network issue. Try Telnet on the IP and port provided by Bloomberg platform. If Telnet fails kindly check the firewall and contact Bloomberg support.
- 2. FIX Session frequently disconnect.
- Ans: It's mostly due to poor network connectivity. Kindly contact Bloomberg support and ask them to confirm session heartbeats.
- **3.** How to enable logs for Bloomberg.
- Ans: Add debug log categories 'BloombergFIT, FIX, FpML' on EngineServer/Navigator.
- 4. Unable to connect to FIX session during certain time.

OR

FIX connectivity stops after a certain period.

Ans: Bloomberg provides StartTime and EndTime, only during which user can connect to FIX session. User will not be able to connect to fix session before StartTime and after EndTime.

This connectivity time frame can be controlled via FIX properties.

Example:

StartTime=07:00:00

EndTime=23:00:00

TimeZone=America/New_York

For StartTime and EndTime value, please contact Bloomberg support.

5. FIX message gets rejected due to missing or unorder tag.

OR

Data-Dictionary outdated / out of sync.

Ans: BloombertFIT module try to keep it data-dictionary in sync with Bloomberg platform. But in case if data-dictionary is outdated and fix messages started getting rejected from FIX session, then user is advised to disable the data-dictionary validation by adding property 'ValidateIncomingMessage=N' in fix config file. Disabling the validation check will stop the fix messages from getting rejected. BloombergFIT module has its own data validation check and if any required data is missing or invalid on fix message, BloombergFIT module will log an error in task-station.



- 6. How to convert PEM certificate to JKS certificate.
- Ans: PEM certificates contains following:
 - a) CAcert.pem
 - b) certificate.pem
 - c) key.pem
 - d) A passphrase over phone to the designated person

To use this information as a Java Key Store (JKS) in our application below steps required. Assuming all commands are run into the same directory where the above files are present.

Step 1: Verify the passphrase for the private key:

openssl rsa -in key.pem -check [when prompted provide the passphrase for the key]

Step 2: With your private key and public certificate, you need to create a PKCS12 keystore first, then convert it into a JKS.

openssl pkcs12 -export -name CAcert.pem -in certificate.pem -inkey key.pem -out <**your_key_identifier_name**>. p12

keytool -importkeystore -destkeystore <**your_key_identifier_name**>.jks -srckeystore <**your_key_identifier_name**>.p12 -srcstoretype pkcs12

Step 3: To verify the contents of the JKS, you can use this command:

keytool -list -v -keystore <your_key_identifier_name>.jks

- 7. How to convert PEM certificate to PKCS12 certificate.
- Ans: PEM certificates contains following:
 - a) CAcert.pem
 - b) certificate.pem
 - c) key.pem
 - d) A passphrase over phone to the designated person

To use this information as a Java Key Store (JKS) in our application below steps required. Assuming all commands are run into the same directory where the above files are present.

Step 1: Verify the passphrase for the private key:

openssl rsa -in key.pem -check [when prompted provide the passphrase for the key]

Step 2: With your private key and public certificate, you need to create a PKCS12 keystore.

openssl pkcs12 -export -name CAcert.pem -in certificate.pem -inkey key.pem -out <**your_key_identifier_name**>. p12