



PONTON
WE ARE THE 2 IN B2B

Hash-based UTI Generation

Version: 86

Date: 16-Feb-2024



Copyright Notice

This document is the confidential and proprietary information of PONTON GmbH ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with PONTON GmbH.

Table of Contents

Copyright Notice	2
Table of Contents	3
1. Hash-based UTI Generation	5
1.1. Version Control	5
1.1.1. Changes in Version 1.2 (06.01.2014)	5
1.1.2. Changes in Version 1.3 (January 29).....	5
1.1.3. Changes in Version 1.4 (12.03.2014)	5
1.1.4. Changes in Version 1.5 (09.11.2023)	6
1.2. What is the Problem?.....	6
1.3. How does the UTI hash function work?	6
1.4. The UTI generator works as follows:	7
1.5. Central or decentral generation of a UTI?.....	8
1.6. Standardisation	9
1.7. Frequently Asked Questions.....	9
2. UTI generation algorithm	15
2.1. Field types.....	15
2.2. Used input fields for hash creation.....	16
2.2.1. General specification for UTI hash input data elements	16
2.3. Algorithm.....	20
3. Example deal	21
3.1. Example deal 1	22
3.2. Example deal 2 with slightly different deal data	23
4. CpML --> UTI-Field-Mappings	24
4.1. Fallback UTI generation in the CMS context	24
4.2. General Information	24
4.3. Commodities	24
4.3.1. Physical Commodity Trades (FOR, OPT, PHYS_INX, OPT_PHYS_INX)	26
4.3.2. Financial Commodity Trades (Financial deals based on a fixed price leg: FXD_SWP, OPT_FXD_SWP, OPT_FIN_INX).....	28

4.3.3. TradeConfirmation (Financial deals with two float legs: FLT_SWP, OPT_FLT_SWP)	29
4.4. FX	31
4.5. IRS	33
4.6. ETDs	35

1. Hash-based UTI Generation

1.1. Version Control

1.1.1. Changes in Version 1.2 (06.01.2014)

- Encoding will be UTF-8
- Whitespaces at the beginning / end of values are not allowed
- Numbers will not have thousands separator (e.g. use 30000000 and not 30,000,000)
- The RunningNumber is created based on equal KeyData

1.1.2. Changes in Version 1.3 (January 29)

- UTI length reduced to 42 characters for length compatibility with USIs under Dodd-Frank
- Rules and value lists adjusted for CpML compatibility
- "Commodity" key element renamed in "Product", Commodity Reference Code renamed in Price/Rate Reference Code, as a generalisation across asset classes
- Extension for FX, IRS, and ETD asset classes
- Use "." for decimal delimiter (e.g. 3000.0000 and not 3000,0000)
- Buyer is selected by the LEI code, not by the EIC code.

1.1.3. Changes in Version 1.4 (12.03.2014)

- Extend generated Running Numbers to 01 to 99 followed by AA, AB, ... ZY, ZZ for trade with the same commercial terms
- UTI generator accept capital letters and numbers as RunningNumber
- Rules and value lists adjusted for:
 - Price for Financial commodity deals with two float legs
 - EffectiveDate and MaturityDate for Physical commodity deals

1.1.4. Changes in Version 1.5 (09.11.2023)

- new implementation adopts the ISO Format ([ISO 23897](#))
 - only uppercase latin letters a to z without modifications and digits
 - length is limited to 52
 - (18 uppercase characters or digits) (2 digits) (up to 32 uppercase characters or digits)
 - Removed UTI-Generator as Excel macro

1.2. What is the Problem?

UTIs need to be generated for trades under EMIR and REMIT for both counterparties to identify their trades by the same ID. This is complicated since the two counterparties may have difficulties to exactly find those two trade data sets which belong together.

Currently, each of the two counterparties can only issue a local trade ID with a different UTI value on each side. Dissemination of the related UTIs of each side is considered as one of the most complicated issue when reporting under EMIR as there is no standard process or best practise how to exchange a UTI between the right traders for related trade data.

Using the EFETnet eCM service would help link the two trade data sets and provide a UTI to buyer and seller but this might be too late as the deadline for reporting is t+1.

One solution would be to use the trade ID issued by an exchange or broker platform but this might also be complicated if a trade is split into two half-trades: the counterparty isn't any more the original one but, instead, the CCP or the clearing bank may slip into that role. And even worse, for some trading organisations it might be too much an effort to implement the import of platform-generated UTIs prior to the go-live of EMIR.

To make it more complicated: How to assign UTIs to historic trades? Or how to transfer UTIs if fax is used to exchange trade confirmations? And do both parties always agree on the same principle to share a UTI? Probably not.

Fortunately, we found a solution that is simple and usable by any type of trading organisation. It is based on using a hash function which allows to create exactly the same UTI value if the input data is the same.

1.3. How does the UTI hash function work?

What does a hash function do? It takes an input text and transforms it into a data value of a given length (e.g. with a length of 30 characters).

The input text could be the data elements taken from the example further down. If put in the right order as a string, the result is the following:

```
5299002Z3I75TD5QSV03SN633FGTWNSOZMOJY6802013-11-11PowerFOR2014-01-012015-01-011000.01001200000.0000EUR
```

This includes the key data of a trade. The output value of the hash function would be: LEI45678901234567890DBBXNGOAZT8QSECEJAJ0AROKU18HQR.

If only a little change is made to the input data, e.g.,

```
5299002Z3I75TD5QSV03SN633FGTWNSOZMOJY6802013-11-11PowerFOR2014-01-012015-01-011000.01001300000.0000EUR
```

the part without the prefix would be completely different, e.g.:

```
LEI456789012345678903DHTZNKUG0ZBYPBYUK4OF5GPNUBC1U .
```

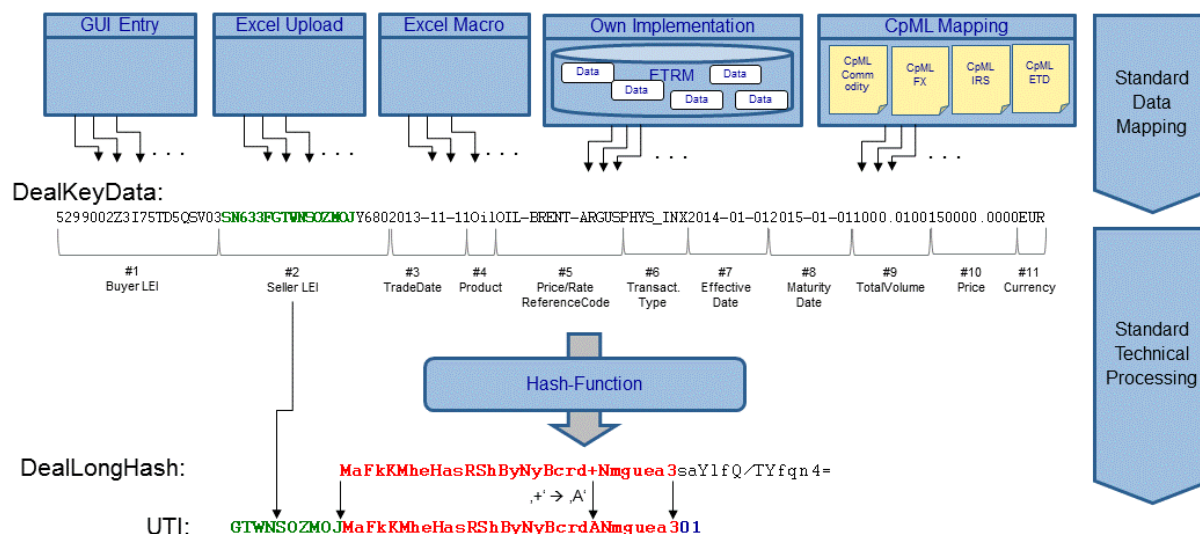
It is therefore extremely unlikely that the hash function will lead to exactly this same value for a second trade with different trade data (less than 1 : 1.000.000.000.000).

If now the two counterparties sit on separate islands in China and in Brasil, respectively, without an Internet access, but knowing the prefix and that the buyer is BP, the seller is Shell, the trade date is 15.11.2013 etc. then they would be in the position to locally create exactly the same hash value.

1.4. The UTI generator works as follows:

- A standard set of key trade data elements is transformed into an initial trade ID hash, e.g. "DBBXNGOAZT8QSECEJAJ0AROKU18HQR". The used data items are: Buyer LEI, Seller LEI, Trade Date, Product, PriceRateReferenceCode, TransactionType, EffectiveDate, MaturityDate, TotalVolume, Price, and Currency.
- If there is more than one trade with the same key data (i.e., trades with the same commercial terms on the same trade date), the ID is made unique by addition of a running number. The two equivalent trades would then be assigned the unique IDs "DBBXNGOAZT8QSECEJAJ0AROKU18HQR01" and "DBBXNGOAZT8QSECEJAJ0AROKU18HQR02".
- Finally, the LEI of the party generating the hash is used as a prefix. LEI45678901234567890DBBXNGOAZT8QSECEJAJ0AROKU18HQR01 and LEI45678901234567890DBBXNGOAZT8QSECEJAJ0AROKU18HQR02 if LEI is "LEI45678901234567890".

The part without the prefix is independent of the party (buyer or seller) generating the UTI values since both parties using the same algorithm will generate matching UTIs independently and without the need to exchange data.



1

Figure 1 UTI generation process

UTI generation works in two steps:

1. Data mapping. This is either done by retrieving the UTI input data out of the ETRM system or out of the CpML documents, or by manually data entry into the UTI Web GUI. Data mapping is in the responsibility of the trader, the rules how to apply correct data elements, how to format and how to order them is defined in section [CpML --> UTI-Field-Mappings](#).
2. Technical processing: This is an algorithm that works identically for all use cases. Is was defined by EFET and developed by Ponton as a Java reference implementation.

1.5. Central or decentral generation of a UTI?

UTIs can be created by different means

- **Central UTI creation GUI:** you may go to <http://uti.equias.org/> and fill out the UTI creation form. This is the easiest way but is has one disadvantage: should there be a second trade on the same day with exactly the same key data the central form-based

generator won't know about this. I.e., it will always create a "...01" UTI value. So this only works for demo purposes or if you know exactly that you are not using this for "clone" trades.

- **"Do it yourself"**: Finally, it makes sense for your IT staff to build the UTI generator into your local trading system such that the UTI is generated on-the-fly whenever you enter a trade. For your inspiration, sample Java code is made available [here](#).

1.6. Standardisation

It is very important that hash-based UTIs are created exactly in the same way everywhere. Although the UTI generator may be used in a test phase for some more days or weeks, we finally have to freeze the algorithm such that discrepancies are avoided. The following rules need to be obeyed by each party for distributed UTI creation:

- A standard set of trade data elements needs to be selected
- Data has to be organised in a standard order
- Data elements have to be formatted in a standard way
- A hash value has to be created using a standard algorithm, further transformations apply in a standard way: transformation of the hash value in BASE64 format and then further transformation in alphanumeric format
- Trades which result in the same hash value consist probably of the same trade data – but to achieve *Unique* Trade Ids they have to be made unique – also in a standard way (by adding the running number)
- The easiest way to accomplish this is to use the same software implementation that validates and processes data in exactly the same way.

1.7. Frequently Asked Questions

- *Is it possible that the same UTI is created for two trades with different key data?*
In principal yes, but the likelihood is extremely low such that this possibility can be ignored. Also, in line with the ISDA protocol, UTIs should be added to the trade confirmation leading to reconciliation of any discrepancies between counterparties.
- *What if there is a mistake in the key data such that our UTI has a different value than the UTI generated by our counterparty?*
This may happen, and it will probably happen in 1-2% of all cases, depending on the point in time when the UTI is generated. If this is detected, e.g., through eCM confirmation matching, any discrepancies should have been remedied. If it takes place before confirmation matching, confirmation matching may lead to a correction

of data in this case the seller's UTI will be disseminated through confirmation (after application of any amendments and upon successful matching) according to the ISDA protocol, this will be the case even if the UTI was generated using trade data which was subject to amendment in confirmation. It should not be assumed therefore that the UTI is equivalent to the original data used to generate it, as the data may be subsequently changed (as a result of confirmation or another lifecycle event resulting in trade detail amendment) whilst the UTI will always remain unchanged following successful confirmation.

- *Is the UTI safe? Can it be reverse engineered?*

The current UTI algorithm uses an open standard describing all the input fields and the hashing algorithm. The list of fields has been chosen balancing the uniqueness requirements versus the easiness to implement the generation process. Using static data for some of the input fields and applying common sense to some other fields could make it possible to apply a brute force reverse engineering mechanism to the UTIs. It remains therefore advisable to safeguard the confidentiality of UTIs and ensure they are only handled and exchanged using appropriately secured communication processes. EFETnet uses such secure channels for both electronic Confirmation Matching (eCM) and electronic Regulatory Reporting (eRR) in all directions: data exchange with traders and brokers as well as with trade repositories.

- *Can I use this technique for backloading?*

Yes, the approach is particularly relevant to resolving back loading issues related to the generation of matching UTIs for historic transactions. Under the assumption that historic trades are confirmed and therefore the likelihood for discrepancies is mostly avoided, you and your counterparty may independently create the same set of UTIs prior to back loading under EMIR and REMIT.

- *What does this service cost?*

Nothing. It is a free service provided by EFETnet.

- *Who is allowed to use this service?*

Everyone. It has been extended for assets classes such as FX and IRS since version 1.3 and would principally also work for others – we only need to define standard values for the Product and PricerateReferenceCode fields.

- *My trade data is very sensitive, how safe is it?*

We process your data on-the-fly without storing anything on disk.

- *Has this been tested by someone?*

Yes, Centrica did some tests with counterparties and with more than 1.000 trades the hit rate (i.e. generating the same UTI value on both sides) was 99.7%.

- *Why do you add the index as the last step and not before hash creation?*
We believe that is an important information for users to recognise if there are clone trades such that they can dedicate special attention to them. On the other hand, this doesn't disclose any sensitive trade details to a third party. And finally, the function that generates UTIs can easily recognise if an UTI has already been created for another clone trade with the same trade data.

- *How many clone trades can be indexed?*
So far, we have implemented indexing between 01 and 99. If there is a request to extend this range, we would continue countinlexically from "AA" to "ZZ" which would provide another 676 clone members.

- *Does the algorithm support formula swaps?*
Yes, since version 1.3 you may use your formula ID instead of the price/rate reference code for any commodity trade that uses float price information leg.

- *What is the risk of a UTI clash if clone trades are executed on different platforms?*
 - Let's assume that a trade is executed bilaterally between two traders X and Y, they would both locally generate a USI **XY01**. Then the same trade is executed on broker B. B issues USI **XY01** because B does not know about the "bilateral XY01".
 - This problem occurs if two traders use different UTI assigning agencies *and* change them within a set of clone trades (jointly or unilaterally).
 - Another example:
 - CP X and CP Y do a trade through broker B, B issues UTI **XY01** for this first trade in a clone set.
 - CP X uses the UTI created by the broker (**XY01**), CP Y creates an own UTI: also **XY01** as this is the first case – everything fine so far!
 - Then a clone trade is created, B issues **XY02** (as B knows about the previously created XY01) and X accepts it from the broker. But now CP Y does neither create an own one nor uses the UTI created by the broker (in the first case Y would see a uniqueness violation which would lead to an increment to XY02 if implemented correctly, in the second case B would increment the UTI anyway).
 - Instead, Y leaves the UTI element empty and lets EFETnet create the UTI through the eCM channel. As EFETnet does not know about the history, EFETnet would create a **XY01** if Y is the seller and would even override X's XY02 with a XY01.

Can this be avoided? Is not realistic that traders change the UTI generation issuer within a day for the same set of clone trades. Therefore, traders MUST NOT change the UTI generating agent within a set of clone trades.

- *What other situations exists which may lead to a UTI clash?*

The following may happen:

- Two traders do exactly the same trade twice. However, trader X makes a mistake when entering the deal data for trade 2, i.e., the following is created
 - Trader X:
 - Deal 1: YXCVBNM01
 - Deal 2: QWERTZU01
 - Trader Y:
 - Deal 1: YXCVBNM01
 - Deal 2: YXCVBNM02

At a later time, trader X corrects the deal data of deal 2 to YXCVBNM01 as the history of UTI creation for deal 1 is not available anymore.

Solution: UTI generating entities MUST keep track of previously created UTIs to detect clashes and to increment the UTI index if a clash occurs.

- *Which value to use for data elements “TotalVolume” and “Price” in case of a physical trade with a list of delivery time intervals?*

- for TotalVolume: calculate for each time interval the delivery hours, summed up across all delivery time intervals.
- For Price use the CpML TotalContractValue, it is calculated as defined in the CpML standard: for each time interval: delivery hours * hourly price, summed up across all delivery time intervals.

- *How is the update process for the UTI generator and for the static data code used?*

- Static data codes, specifically for price / rate reference code, are maintained by EFET and published here: <http://staticdata.efet.org/view.aspx?d=IndexCommodity>. Whenever we take over such extensions of static data, we will publish this in the UTI Generator change log. Any such updates apply for all UTI Generator channels (Web, Java reference code) at the same time to avoid inconsistencies.
- Code updates / algorithm updates: The same applies here, we will announce an update date and adjust all channels at the same time.

- Please send in your request for new static data to support@efetnet.org. The support team will analyse the request, ensure compatibility and compliance with existing static data and post it onto the static data webpage
- *How to cope with ESMA's [Q&A update](#) of February 11th 2014, which partly also affects the UTI format and generation process?*
 - As this is obviously too late to be addressed for go-live under EMIR, we suggest the following: for the time being, keep the UTI algorithm and format as it stand today. EFETnet will will publish an impact analysis based on the latest FAQs on the UTI generation and dissemination next week and then organise a conf call to discuss potential way(s) forward, including implementation timing and change management. After having discussed with ESMA how to further proceed here, EFETnet will announce an update date well in advance such that all users of the UTI Generator and Ponton can synchronise their switch-over to a new schema.
- *How to transfer a UTI to the CMS for eCM and how is a created UTI transfered back to the participant?*
 - Trader --> eCM: The UTI is transferred to the eCM service using the CpML ECMLEnvelope/EUReporting section.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ECMLEnvelope
  xsi:noNamespaceSchemaLocation=http://www.efet.org/schemas/V4R2/EFET-ENV-V4R2.xsd
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <TradeConfirmation>
    <DocumentID>CNF_123@example.com</DocumentID>
    <!-- content of a wellformend CNF -->
  </TradeConfirmation>
  <ECMLAdditionalData>

  <ReferencedDocumentID>CNF_123@example.com</ReferencedDocumentID>

    <CreationTimestamp>...</CreationTimestamp>
    <EUReporting>
      <UTI>...</UTI>
    </EUReporting>
  </ECMLAdditionalData>
</ECMLEnvelope>

```

- eCM --> Trader: A UTI generated by the eCM or eRR service is sent back to the trader, using the eRR BoxResult:

```

<BoxResult>
  <DocumentType>CNF</DocumentType>
  <DocumentID>CNF_123@example.com</DocumentID>
  <DocumentVersion>1</DocumentVersion>
  <ebXMLMessageId>CENTRAL_MATCHING</ebXMLMessageId>
  <State>MATCHED</State>
  <Timestamp>2013-11-12T10:29:27</Timestamp>
  <Counterparty>
    <DocumentID>CNF_123@other.com</DocumentID>
    <DocumentVersion>1</DocumentVersion>
    <ebXMLMessageId>CENTRAL_MATCHING</ebXMLMessageId>
  </Counterparty>
  <EUReporting>
    <UTI>AYE8YOZQ4HXXXXXXXXXX</UTI>
  </EUReporting>
</BoxResult>

```

- To use the abovementioned Envelope and eRR BoxResult with <EUReporting>, CMS users have to apply this patch for their **UAT Ponton X/P installation**:

<http://ponton-consulting.de/downloads/xp/efet/hub/efet-dist-3.1-cms-test-ECM42-patch.zip>

- How to transfer a UTI to the CMS for eRR and back to the participant?*
 - In case of eRR, the UTI is part of the reporting envelope: CpmIDocment/Reporting/Europe/EURegulatoryDetails/UTI
 - An eRR BoxResult providing a UTI back to the participant looks as follows:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ERRBoxResult>
  <DocumentID>CPML_20131002_tradeid0001@party</DocumentID>
  <DocumentVersion>1</DocumentVersion>
  <Timestamp>2013-10-15T15:37:07.868+01:00</Timestamp>
  <EuropeResult>
    <Action>REPORTING</Action>
    <Result>OK</Result>
    <Regime>Emir</Regime>
    <Repository>UNAVISTA</Repository>
    <ReportingResult>
      <TradeID>tradeid0001</TradeID>
      <UTI>PARTY000000000000001</UTI>
    </ReportingResult>
  </EuropeResult>
</ERRBoxResult>

```

2. UTI generation algorithm

2.1. Field types

Type	Description	Regular expression
String	A UTF-8 representation of a value. No whitespaces characters are allowed at the beginning or end of the value. For whitespace definition refer to the REGEX specification.	<code>[^\s] (. * [^\s]) ?</code>
Date	A date is a String with the following format YYYY-MM-DD. Where <ul style="list-style-type: none"> • YYYY is the year of the date • MM is the month of the date (01-12) • DD the day of the month of the date (01-31) 	<code>\d\d\d\d-\d\d-\d\d</code>
Enum	A enumeration is a String with predefined values. Please refer to the field description for allowed values.	
Decimal	A decimal is a String with <ul style="list-style-type: none"> • always with 4 decimal precision, e.g. (10000.0000 or 0.0000 or 0.1000 or 0.0001) • see regular expression 	<code>\d+\.\d{4}</code>
RunningNumber	The number to make trade identifiers of trades with identical commercial terms unique	<code>[A-Z0-9] [A-Z0-9]</code>

2.2. Used input fields for hash creation

If the buyer is that party with a lexically lower party code (applicable in case of swaps with two float legs), parties are ordered based on their LEI code, not the EIC code. For the avoidance of doubt: A conversion EIC code --> LEI code has to be performed prior to UTI generation.

The following Table defines the order, data types and content types that are generally used for the respective hash input data values.

Depending on the combination of TransactionTypes and asset classes the following trade types are defined:

1. physical commodity trades
2. Index-based trades
3. Fixed-float swaps / swaptions
4. Float-float swaps / swaptions
5. FX trades
6. IRS trades
7. ETD trades

Trade type specific data mappings are defined in Section [CpML --> UTI-Field-Mappings](#))

2.2.1. General specification for UTI hash input data elements

Field name	Order	Usage	Type	Description
BuyerID	1	Mandatory	String	Enter here the LEI of the buyer. It will be used to generate the UTI value, but not for the namespace.
SellerID	2	Mandatory	String	Enter here the LEI of the seller. It will be used for both the UTI namespace and the value. Out of the seller LEI code characters 7-16 are used as the UTI namespace, following the ISDA rule. If there is no seller to the trade (e.g. in case of a swap), the seller of the fixed leg or the counterparty with the lexically lower LEI in a float/float swap goes here.
TradeDate	3	Mandatory	Date	The trade date.

Field name	Order	Usage	Type	Description
Product	4	Conditional	String	<p>For physical commodities: Power, Gas, Oil, Coal, Bullion, Metal, Agriculturals, Paper, ReactivePower, EUAPhase_1, EUAPhase_2, EUAPhase_3, CER</p> <p>For FX trades: FXSpot, FXForward, FXSwap, FXOption, FXForward_Non_Deliverable, FXOption_Non_Deliverable,</p> <p>For IRS trades: IRSwap, Basis, CrossCurrency, Commodity, ForeignExchange, Equity, EquityBond, InterestRate, Credit, InterestRateODRF</p> <p>Find a detailed specification of the CpML data mapping here: CpML --> UTI-Field-Mappings</p>
PriceRateReferenceCode	5	Conditional	String	<p>For financial commodity trades (FLT_SWP, FXD_FXD_SWP, FXD_SWP, OPT_FIN_INX , OPT_FLT_SWP, OPT_FXD_SWP.) and for physical commodity trades based on a price index (PHYS_INX, OPT_PHYS_INX)</p> <ul style="list-style-type: none"> • use one of the index references (attribute description of the EFET element) as defined in the eCM standards for XML type "ISDACommodity-DefinitionsType". E.g., "OIL-BRENT-IPE". • These allowed static data values as published by EFET are published here: http://www.efet.org/Standardisation/Static-data (attribute code of the EFET element in the XML representation). <p>Otherwise this field is left blank (for FOR and OPT_FOR)</p> <p>Find a detailed specification of the CpML data mapping in section CpML --> UTI-Field-Mappings</p>
TransactionType	6	Mandatory	Enum	<p>Use one of the following: DAH, IND, SPT, FOR, FUT, PHYS_INX, OPT_PHYS_INX, FXD_SWP, FXD_FXD_SWP, FLT_SWP, OPT, OPT_FXD_SWP, OPT_FLT_SWP, OPT_FIN_INX. Detailed rules</p>

Field name	Order	Usage	Type	Description
				when to use which TransactionType value are given here: CpML --> UTI-Field-Mappings .
EffectiveDate	7	Conditional	Date	This is the start date in case of a physical delivery schedule or the effective date in case of a financial product.
MaturityDate	8	Conditional	Date	This is the end date in case of a physical delivery schedule or the maturity date in case of a financial product.
TotalVolume	9	Mandatory	Decimal	<ul style="list-style-type: none"> 4 decimal places required precision to normalise data entries. This is the overall delivery volume in case of physical trades. The assumption is that both counterparties have agreed on the same unit of measure. TotalVolume (QuantityType): Always use positive quantities without a sign. If a quantity has more than 4 decimal places, truncate any further decimal places to 4. (e.g.: 1000.12345678 --> 1000.1234 or 1000.666666 --> 1000.6666) <p>Find a detailed specification of the CpML data mapping in section CpML --> UTI-Field-Mappings</p>
Price	10	Conditional	Decimal	<ul style="list-style-type: none"> Use TotalContractValue or unit price, depending on trade type. TotalContractValue (PriceType): Always use positive quantities without a sign. If a quantity has more than 4 decimal places, truncate any further decimal places to 4. (e.g.: 1000.12345678 --> 1000.1234 or 1000.666666 --> 1000.6666) <p>Find a detailed specification of the CpML data mapping in section CpML --> UTI-Field-Mappings</p>
Currency	11	Conditional	String	The currency of the price of the deal, based on

Field name	Order	Usage	Type	Description
				the ISO 3-letter code (ISO 4217) Find a detailed specification of the CpML data mapping in section CpML --> UTI-Field-Mappings

2.3. Algorithm

Step	Description	Name
1.	The fields will be concatenated to the <code>DealKeyData</code> . Optional fields will be left out.	<code>DealKeyData</code>
2.	The <code>DealKeyData</code> will be hashed with the SHA-256 algorithm to a <code>DealRawHash</code> which has a size of 256 bits.	<code>DealRawHash</code>
3.	<ol style="list-style-type: none"> 1. The <code>DealRawHash</code> will be encoded by Base64 (RFC 4648) into the <code>DealLongHash</code>. 2. The character '+' will be replaced by 'A' and the character '/' will be replaced by 'B' in <code>DealLongHash</code>. 3. The first 30 characters of the result of the replacement step (step 3.2.) will be used as <code>DealHash</code>. 	<code>DealHash</code>
4.	<p>The <code>DealHash</code> will be made unique by addition of a <code>RunningNumber</code> starting with 01.</p> <p>If there are more than one trades with the same <code>KeyDealData</code> (same deal hash so far) (i.e., trades with the same commercial terms on the same trade date), the next <code>RunningNumber</code> will be 02 to 99 followed by AA, AB, ... ZY, ZZ. This will give us 775 unique numbers.</p>	<code>RunningNumber</code>
5.	An LEI will be used as <code>UTIPrefix</code>	<code>UTIPrefix</code>
6.	<p>Finally concatenate the UTI:</p> $\text{UTI} = \text{UTIPrefix} + \text{DealHash} + \text{RunningNumber}$	UTI

3. Example deal

3.1. Example deal 1

Field name	Value
LEI	LEI45678901234567890
Buyer	5299002Z3I75TD5QSV03
Seller	SN633FGTWNSOZMOJY680
Trade Date	2013-11-11
Product	Power
Price/Rate ReferenceCode	
Transaction Type	FOR
Effective Date	2014-01-01
Maturity Date	2015-01-01
Total Volume	1000.0100
Price	1200000.0000
Currency	EUR
RunningNumber	01
DealKeyData	5299002Z3I75TD5QSV03SN633FGTWNSOZMOJY6802013-11-11PowerFOR2014-01-012015-01-011000.01001200000.0000EUR
DealLongHash	d//XNGOazt8QSeCEJaj0ARoKU18HqrifbSLLNqfcRhA=
UTI	LEI45678901234567890DBBXNGOAZT8QSECEJAJ0AROKU18HQR

3.2. Example deal 2 with slightly different deal data

Field name	Value
LEI	LEI45678901234567890
Buyer	5299002Z3I75TD5QSV03
Seller	SN633FGTWNSOZMOJY680
Trade Date	2013-11-11
Product	Power
PriceRate ReferenceCode	
Transaction Type	FOR
Effective Date	2014-01-01
Maturity Date	2015-01-01
Total Volume	1000.0100
Price	1300000.0000
Currency	EUR
RunningNumber	01
DealKeyData	5299002Z3I75TD5QSV03SN633FGTWNSOZMOJY6802013-11-11PowerFOR2014-01-012015-01-011000.01001300000.0000EUR
DealLongHash	3dhTznkug0zByPByUk4oF5gPNU/C1umMdOpVFhx7reQ=
UTI	LEI456789012345678903DHTZNKUG0ZBYPBYUK4OF5GPNUBC1U

4. CpML --> UTI-Field-Mappings

4.1. Fallback UTI generation in the CMS context

The following algorithm will be used if the hash base UTI generation can not be performed on the given trade in the CMS system, e.g. one of the party could not be mapped to a LEI code or it has not even an LEI code (parties from outside the EU)

Concatenate the following parts:

- The characters 7-16 (6WMSOHHJW5) of the EFETnet LEI ([5493006WMSOHHJW5ZO63](#))
- The time stamp the UTI is generated in format yyyyMMddHHmmssSSS
 - yyyy (the year), MM (the month starting with 01), dd (the day starting with 01), HH (the hour 0-23), mm (the minutes), ss (the seconds), SSS (the milliseconds)
- 15 random alphanumeric chars consisting of numbers 0-9 and characters A-Z excluding the letters 'I' and 'O'

4.2. General Information

The following information provides a precise definition how to map UTI key data from CpML document input. This applies to the asset classes Commodities, FX, IRS, and ETDs. Information on CpML can be found here: www.cpml.eu. Intermediate releases of CpML schemas and documentation can be found here: <https://err.ponton-consulting.de/> (password required).

The entry points for asset class specific XML paths are here:

- Commodities: CpmlDocument/TradeConfirmation/...
- FX: CpmlDocument/FXTradeDetails/...
- IRS: CpmlDocument/IRSTradeDetails/...
- ETDs: CpmlDocument/ETDTradeDetails/...

4.3. Commodities

Remarks

- Data element name "Commodity" in version 1.2 is replaced by "Product" in version 1.3 as the scope of UTI generation also covers other asset classes than commodities
- Data element name "CommodityReferenceCode" in version 1.2 is replaced by "PriceRateReferenceCode" in version 1.3 for the same reason

4.3.1. Physical Commodity Trades (FOR, OPT, PHYS_INX, OPT_PHYS_INX)

Field	Mapping (source: TradeConfirmation/...)	Remark
Buyer	BuyerParty	Use the buyer LEI here
Seller	SellerParty	Use the seller LEI here
TradeDate	TradeDate	In format: YYYY-MM-DD
Product	Commodity	Source values are of type EnergyProductType with the following value "Power", "Gas", "Oil", "Coal", "Bullion", "Metal", "Agriculturals", "Paper", "ReactivePower", "EUAPhase_1", "EUAPhase_2", "EUAPhase_3", "CER"
PriceRateReferenceCode	FOR, OPT: leave blank PHYS_INX, OPT_PHYS_INX: FloatPriceInformation/FormulaID if present, else FloatPriceInformation[1]/CommodityReferences/ CommodityReference[1]/CommodityReferencePrice	Applies to index-priced trades only. Use the first occurrence of CommodityReferencePrice (which is the ISDA commodity definition). Formula trades use the FormulaID instead.
TransactionType	TransactionType	FOR, OPT, PHYS_INX, OPT_PHYS_INX
EffectiveDate	TimeIntervalQuantity[1]/DeliveryStartDateAndTime if present, else EUATradeDetails/EmissionsDeliveryDate	Get the date part of DeliveryStartDateAndTime from the first TimeIntervalQuantity Emission trades use the EmissionsDeliveryDate. Format: YYYY-MM-DD
MaturityDate	TimeIntervalQuantity[last]/DeliveryEndDateAndTime if present, else EUATradeDetails/EmissionsDeliveryDate	Get the date part of DeliveryEndDateAndTime from the last TimeIntervalQuantity

Field	Mapping (source: TradeConfirmation/...)	Remark
	e	Emission trades use the EmissionsDeliveryDate. Format: YYYY-MM-DD
TotalVolume	TotalVolume	CNF it is dot separated and has 8 decimal places. Truncate to 4 decimal places if necessary
Price	FOR: TotalContractValue OPT: OptionDetails/TotalPremiumValue PHYS_INX: leave blank OPT_PHYS_INX: leave blank	In case of a FOR use TotalContractValue (price per unit is not used here as the total contract value is considered more selective, moreover, Price is per TimeInterval so that it may vary from interval to interval) In case of an OPT use OptionDetails/TotalPremiumValue
Currency	Currency	Ignore the "UseFractionUnit" attribute

4.3.2. Financial Commodity Trades (Financial deals based on a fixed price leg: FXD_SWP, OPT_FXD_SWP, OPT_FIN_INX)

Field	Mapping (source: TradeConfirmation/...)	Remark
Buyer	see "Physical Commodity Trades"	
Seller	see "Physical Commodity Trades"	
TradeDate	see "Physical Commodity Trades"	
Product	leave blank	
PriceRateReferenceCode	FloatPriceInformation/FormulaID if present, else FloatPriceInformation[1]/CommodityReferences/ CommodityReference[1]/CommodityReferencePrice	as for index-priced physical trades
TransactionType	see "Physical Commodity Trades"	FXD_SWP, OPT_FXD_SWP, OPT_FIN_INX
EffectiveDate	EffectiveDate	
MaturityDate	TerminationDate	
TotalVolume	see "Physical Commodity Trades"	
Price	FixedPrice	Use FixedPrice in first occurrence of DeliveryPeriods/DeliveryPeriod section
Currency	see "Physical Commodity Trades"	

4.3.3. TradeConfirmation (Financial deals with two float legs: FLT_SWP, OPT_FLT_SWP)

Field	Mapping (source: TradeConfirmation/...)	Remark
Buyer	BuyerParty	Use the buyer LEI For FLT_SWP use greater LEI (BuyerParty, SellerParty) after ascending alphanumeric sorting.
Seller	SellerParty	Use the seller LEI For FLT_SWP use lower LEI (BuyerParty, SellerParty) after ascending alphanumeric sorting.
TradeDate	TradeDate	In format: YYYY-MM-DD
Product	Leave blank	
PriceRateReferenceCode	FloatPriceInformation/ FormulaIDFloatPriceInformation/ CommodityReferences/ CommodityReference/ CommodityReferencePrice	Use the first leg (with lower LEI of FloatPricePayer after alphanumeric sorting). If this is a formula leg, use the formula ID, otherwise use TradeConfirmation/FloatPriceInformation/CommodityReferences/CommodityReference/CommodityReferencePrice (Use the first CommodityReference). Type is ISDACommodityDefinitionsType
TransactionType	TransactionType	Use values: FLT_SWP, OPT_FLT_SWP
EffectiveDate	EffectiveDate	Use EffectiveDate
MaturityDate	TerminationDate	Use TerminationDate
TotalVolume	TotalVolume	CNF it is dot separated and has 8 decimal places ==> Truncate to 4 decimal places if necessary, must be converted to dot separated and 4 places
Price	FloatPriceInformation/ FormulaSpreadInformation/ SpreadInformation/ SpreadAmount or	<u>Formula Swap</u> (i.e. FloatPriceInformation[1]/FormulaID is present) <ul style="list-style-type: none"> use the first occurrence FloatPriceInformation[i]/FormulaSpreadInformation/S

Field	Mapping (source: TradeConfirmation/...)	Remark
	FloatPriceInformation/ CommodityReferences/ CommodityReference/ SpreadInformation/ SpreadAmount.	<p>preadAmount</p> <ul style="list-style-type: none"> if none found, use 0.0000 (numerical zero) <p><u>Swap without formula</u> (i.e. FloatPriceInformation[1]/FormulaID is not present)</p> <ul style="list-style-type: none"> use the first occurrence of FloatPriceInformation[i]/CommodityReferences/CommodityReference[j]/SpreadInformation/SpreadAmount if none found, use 0.0000 (numerical zero)
Currency	Currency	

4.4. FX

Field	Mapping (source: FXTradeDetails/...)	Remark
Buyer	BuyerParty	Use the buyer LEI here
Seller	SellerParty	Use the seller LEI here
TradeDate	TradeDate	In format: YYYY-MM-DD
Product	FXProduct	Use: FXProduct, values are: "FXSpot" "FXForward" "FXSwap" "FXOption" "FXForward_Non_Deliverable" "FXOption_Non_Deliverable"
PriceRateReferenceCode	Leave blank	
TransactionType	FXTransactionType	Use FXTransactionType: ("FOR", "OPT", "FXD_FXD_SWP", "OPT_FXD_FXD_SWP", "SPT")
EffectiveDate	FXSingleLeg/ExchangedCurrency/ValueDate or FXOption/EffectiveDate	In case of a "FOR" and "SPT": Use FXSingleLeg[1]/ValueDate (this equals the maturity date) In case of a Swap ("FXD_FXD_SWP"): use first occurrence of ValueDate (chronological order), use chronologically first Value date within the two occurrences of ExchangeCurrency. In case of an option ("OPT", "OPT_FXD_FXD_SWP"): use FXOption/EffectiveDate
MaturityDate	FXSingleLeg/ExchangedCurrency/ValueDate or FXOption/FXExerciseSchedule/ExpiryDate or FXOption/FXExerciseSchedule/ExpiryDateAndTime	In case of a "FOR" and "SPT": Use FXSingleLeg[1]/ValueDate (this equals the effective date) In case of a Swap ("FXD_FXD_SWP"): use last occurrence (chronological order) of FXSingleLeg/ValueDate, use chronologically last Value date within the two occurrences of ExchangeCurrency.


Field	Mapping (source: FXTradeDetails/...)	Remark
		In case of an option ("OPT", "OPT_FXD_FXD_SWP"): extract the date value out of FXOption/FXExerciseSchedule/ExpiryDate or FXOption/FXExerciseSchedule/ExpiryDateAndTime, whichever is populated.
TotalVolume	FXSingleLeg/ ExchangedCurrency/ PaymentAmount or FXOption/ PutCurrencyAmount/ Amount	In case of a "FOR", "SPT" and "FXD_FXD_SWP": Use the first occurrence of FXSingleLeg, then use that occurrence of ExchangedCurrency where BuyerParty = PayerParty, then use PaymentAmount. In case of an option ("OPT", "OPT_FXD_FXD_SWP"): Use the Put amount. CNF it is dot separated and has 8 decimal places ==> Truncate to 4 decimal places if necessary
Price	FXSingleLeg/ ExchangedRate/ SpotRate or FXOption/ PremiumPayments/ PremiumPayment/ PremiumPaymentValue	In case of FOR, SPT, FXD_FXD_SWP: Within the first occurrence of FXSingleLeg, use first occurrence of ExchangedRate/SpotRate. (? Not clarified yet: Should we add ForwardPoints here as done for EMIR reporting?) In case of OPT or OPT_FXD_FXD_SWP use first occurrence of FXOption/PremiumPayments/PremiumPayment/PremiumPaymentValue
Currency	FXSingleLeg/ ExchangedCurrency/ PaymentCurrency or FXOption/ PutCurrencyAmount/ Currency	In case of a "FOR", "SPT" and "FXD_FXD_SWP": Use the first occurrence of FXSingleLeg, then use that occurrence of ExchangedCurrency where BuyerParty = PayerParty, then use PaymentCurrency. In case of an option ("OPT", "OPT_FXD_FXD_SWP"): Use the Put currency.


4.5. IRS

Field	Mapping (source: IRSTradeDetails/...)	Remark
Buyer	BuyerParty	Use the buyer LEI here
Seller	SellerParty	Use the seller LEI here
TradeDate	TradeDate	In format: YYYY-MM-DD
Product	IRSProduct	Use IRSProduct, values are: "IRSwap", "Basis", "CrossCurrency"
PriceRateReferenceCode	SwapStreams/ SwapStream/ CalculationPeriodAmount/ Calculation/ FloatingRateCalculation/ FloatingRateIndex	Use FloatingRateIndex of first float leg if FloatingRateCalculation exists (if there are two SwapStream legs, the first one is where BuyerParty = PayerParty). Otherwise leave it empty
TransactionType	TransactionType	Use TransactionType "FXD_SWP", "FXD_FXD_SWP", "FLT_SWP", "OPT_FXD_SWP", "OPT_FXD_FXD_SWP", "OPT_FLT_SWP"
EffectiveDate	SwapStreams/ SwapStream/ CalculationPeriodDates/ EffectiveDate/ EffectiveDate	Use firstOccurance of SwapStream where PayerParty = BuyerParty
MaturityDate	SwapStreams/ SwapStream/ CalculationPeriodDates/ TerminationDate/ TerminationDate	Use firstOccurance of SwapStream where PayerParty = BuyerParty
TotalVolume	SwapStreams/ SwapStream/ CalculationPeriodAmount/ Calculation/ NotionalSchedule/ NotionalStepSchedule/ InitialValue	Use firstOccurance of SwapStream where PayerParty = BuyerParty CNF it is dot separated and has 8 decimal places ==> Truncate to 4 decimal places if necessary
Price	SwapStreams/ CalculationPeriodAmount/	Use InitialValue of first leg if FixedRateSchedule exists (if

Field	Mapping (source: IRSTradeDetails/...)	Remark
	Calculation/ FixedRateSchedule/ InitialValue	there are two SwapStream legs, the first one is where BuyerParty = PayerParty). Otherwise leave it empty
Currency	Currency	Currency

4.6. ETDs

Field	Mapping (source: ETDTradeDetails/...)	Remark
Buyer	ETDTradeDetails/ BuyerDetails/ BuyerParty or ETDTradeDetails/ ReceiverID	Use the buyer LEI here
Seller	ETDTradeDetails/ SellerDetails/ SellerParty or ETDTradeDetails/ ReceiverID	Use the seller LEI here
TradeDate	ETDTradeDetails/ MTFDetails/ ExecutionTimesta mp or ETDTradeDetails/ BuyerDetails/ ExecutionTimesta mp or ETDTradeDetails/ SellerDetails/ ExecutionTimesta mp	Extract date out of DateTime string
Product	PrimaryAssetClas s	Use PrimaryAssetClass
PriceRateRefere nceCode	CpMLDocument/ Reporting/ Europe/ EURegulatoryDeta ils/ TradeID	Use CpMLDocument/Reporting/Europe/EURegulatoryDet ails/TradeID ( Open issue: We would like to use the platform's product ID here (like, e.g., "FB01" at EEX or "PM4TC CAL11 C40000" in case of NOS), but we found out that some trading organisations do not keep track of these product type and use internal product IDs instead. The alternative would be to use the 'TransactionReferenceNumber' of the

Field	Mapping (source: ETDTradeDetails/...)	Remark
		venue if this is the same for both sides.)
TransactionType	ETDTradeDetails/ ETDTransactionType	Use ETDTransactionType: “FOR”: Physical Forward that settles against a fixed price “OPT”: Option on a physical forward “PHYS_INX”: Physical forward that settles against an index “OPT_PHYS_INX”: Option on a physical forward that settles against an index “FXD_SWP”: Fixed/float swap “FXD_FXD_SWP”: Fixed/fixed swap “FLT_SWP”: Float/float swap “OPT_FXD_SWP”: Fixed/float swaption “OPT_FXD_FXD_SWP”: Fixed/fixed swaption “OPT_FLT_SWP”: Float/float swaption “OPT_FIN_INX”: Option on an index. “FUT”: Exchange traded future (can be traded off exchange but under the terms of the Regulated Market) “OPT_FUT”: Exchange traded option (can be traded off exchange but under the terms of the Regulated Market) “SPT”: Spot transaction.
EffectiveDate	ETDTradeDetails/ ClearingParameters/ Product/ DeliveryPeriod/ DeliveryStartDateAndTime	If it exists, use Date of ClearingParameters/Product/DeliveryPeriod/DeliveryStartDateAndTime Otherwise leave empty.
MaturityDate	ETDTradeDetails/ ClearingParameters/ Product/ DeliveryPeriod/ DeliveryEndDateAndTime	If it exists, use Date of ClearingParameters/Product/DeliveryPeriod/DeliveryEndDateAndTime Otherwise leave empty.
TotalVolume	ETDTradeDetails/ ClearingParameters/ Lots	Use ClearingParameters/Lots.  ERR-980 - ETD UTI calculation: TotalVolumen must match regular expression of '\d+\.\d{4}'. [value=10000] but there are <Lots> and no <TotalVolume> GESCHLOSSEN

Field	Mapping (source: ETDTradeDetails/...)	Remark
		Extend to 4 decimal places.
Price	ETDTradeDetails/ ClearingParameters/ UnitPrice	In case of OPT_* transactions types ClearingParameters/UnitPrice (in case of options this element holds the premium amount) In case of other transaction types the price per unit goes here (also use ClearingParameters/UnitPrice).
Currency	Leave blank	

PONTON GmbH

Dorotheenstraße 64
22301 Hamburg
GERMANY

www.ponton.de