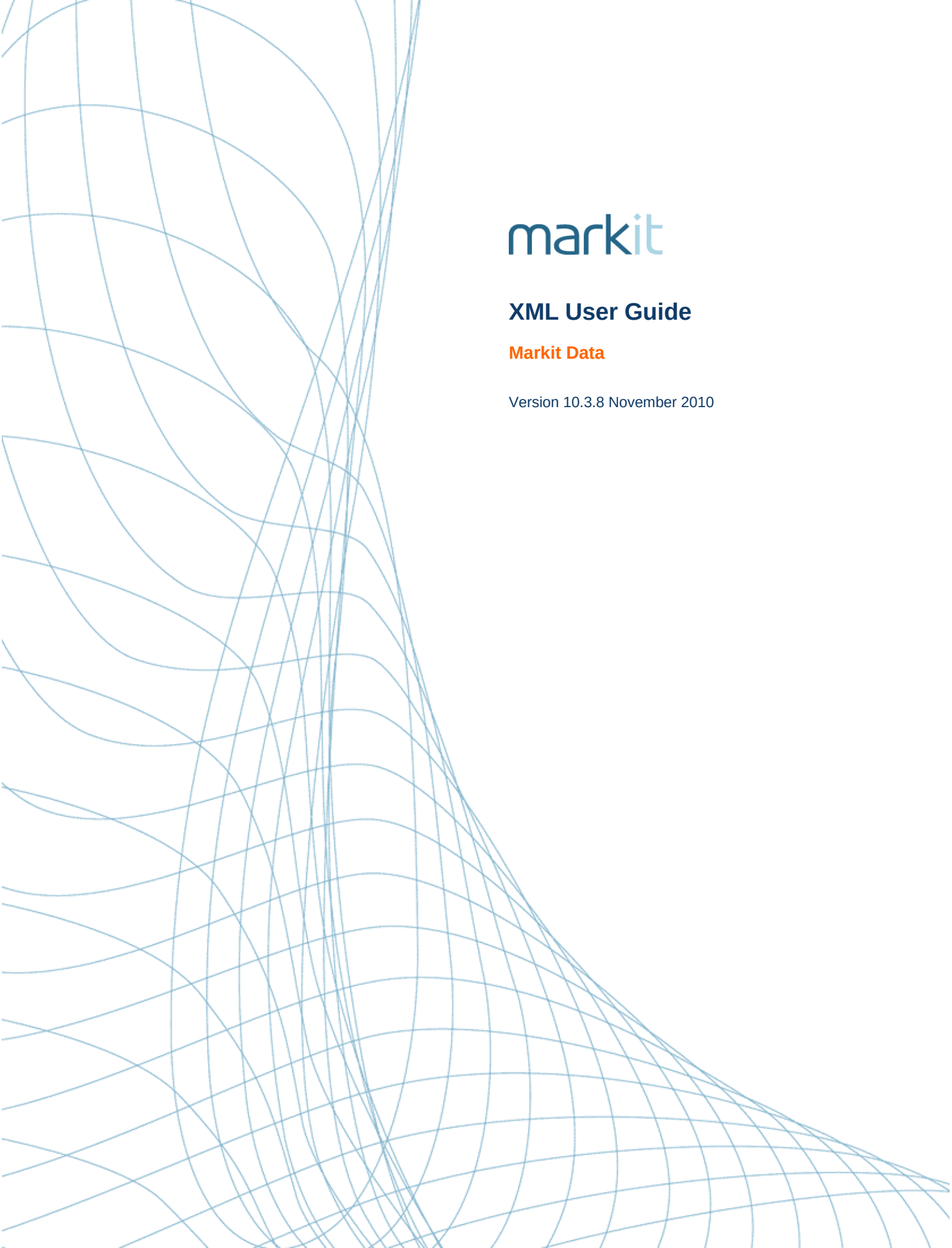


XML User Guide

Markit Data

Version 10.3.8 November 2010

Copyright

Table of Contents

Introduction

This document describes how to:

- Construct an XML file for those customers that are contributing data to the Markit website.
- Retrieve data from Markit's website using the download facilities.

Please note that Markit has the right to modify the format of the XML feeds outlined in this document. Markit will provide notification of any changes and will run the old and new structures in parallel before phasing out the old structure.

If you encounter problems or at any time need additional support, please contact us using the details in the Contact Us section.

Background Information

Markit was founded in 2001 as the first independent source of credit derivative pricing. Today, our data, valuations and trade processing services are regarded as the market standard in the global financial markets, helping our clients to reduce risk and improve operational efficiency.

Composite pricing for the credit products is compiled from contributions made by the major players in the credit markets. This dataset covers single-name credit default swaps, indices (iTraxx, CDX, ABX, LCDX, etc), and bonds.

There are many applications for such a dataset, including: price testing; front-office research; market risk; valuations; and collateral management.

Each contributor to Markit is committed -- on a daily basis -- to provide pricing data from all its books of record and feeds to automated trading systems, as well as other sources of accurate pricing. For a book of record system, this means the daily closing price as recorded for that security or derivative in that system; for automated trading systems it means the last price fed to that system by the trading desk.

Each contributor needs to build a feed to Markit to ensure that a comprehensive daily supply of this data is maintained. The data items are outlined fully later in this document. In summary, the requirements are:

- Bonds
    - Identifier (CUSIP/ISIN, etc)
    - Price
    - Timestamp
    - Position information (optional)
- Default Swaps
    - Identifier (entity code, tier of debt, currency and documentation clause)
    - Default swap curve
    - Position information (optional)

Overview of Steps to Build an XML Feed

The steps involved in connecting a customer XML feed to Markit are as follows:

1. **Customer Details** - The Markit database is initialized with customer details.
2. **Agreeing to the Scope and System Names** - The list of customer trading businesses and systems that will provide data to Markit is agreed and their names entered into the Markit database.
3. **Constructing the XML File** - The customer develops their XML file for transmission to the Markit database. This section describes how to construct an XML file for default swaps, bonds, and convertible bonds.
4. **Creating a User with the Ability to Upload** - An administrator login is provided to the customer. This login enables the customer to administer user accounts. The customer must set up a user with the privileges to upload XML files to the website.
5. **Uploading the XML File Via the Web** - The contributor's XML file is transmitted to the Markit database. Initially, a number of errors display on the upload screen caused by unknown default swap entity codes and prices for bonds that Markit does not hold in our database. Missing entity codes are handled by the mapping logic in the following step. Bond details not held in the Markit database will be obtained by Markit within a few business days. If erroneous data is transmitted to Markit, the errors can be corrected by simply re-submitting the corrected XML file.
6. **Mapping Customer Entity Codes to Markit Entity Codes for CDS** - This step is required only for default swap feeds and involves mapping customer entity tickers to Markit entity tickers with the Markit Entity Mapping screen.
7. **Automating the XML File Transmission to Markit** - The customer can automate the transfer of the XML file via a computer program and examples of these programs are provided.
8. **Retrieving Data Back from the Markit Website** - Once the Markit processes are complete, composite, theoretical, and other contributor data that is quorate is available for download from the Markit website.

In this guide, we will follow the steps outlined above for an imaginary customer who is feeding information from a system called 'Global'.

Customer Details

For security reasons, Markit needs to know the IP address from which the customer will access the website, i.e. the IP address of the customer's Internet facing routers (go to www.whatsmyip.org and tell us the IP address displayed by this website; we will try to work out the address range of your routers).

This security measure safeguards against users who have recently changed jobs accessing the website with their old account details, but from their new employer's network.

Once the customer has informed Markit, Markit will install an administration account for the customer which the customer can use to create a hierarchy of users.

The IP address has not been set up correctly if you receive the following error message when logging into the system: "You are not authorized to log in from this computer (157.190.10.2). Contact Markit to obtain authorization."

The IP address in this error message will be for one of the customer's routers.

Agreeing to the Scope and System Names

The first step is to agree with Markit the trading businesses within the customer's organization that trade credit instruments (all corporate, financial, and sovereign bonds and credit default swaps) and comprise the dataset that the customer is legally obliged to supply Markit. Once this list has been established, it should be easy to define a list of systems (books of record and feeds to automated trading systems) that will feed the Markit database.

The system name helps the customer organize their feeds. If errors or problems occur, the customer and Markit can use the system name to refer to the feed causing the problem.

In the example used in this guide, the customer informs Markit that they have a system called 'Global'.

Since a system may have more than one feed (e.g. from different time zones), each feed is assigned a unique numeric file ID – usually, the file ID of 1.

Constructing an XML File

We now look at building an XML file of bond and default swap data. Below are several tips to remember when constructing an XML file:

- The XML tags are case sensitive and all Markit tags are in lowercase. If a tag is presented in uppercase, you will receive an unknown element error.
- The elements must appear in the sequence described.
- The maximum file size is 2MB. If this file size is exceeded, you will receive the following error message: 'Cannot find file. Check the file exists and does not exceed 2MB in size'. The solution to this problem is to split the file among several file ids.
- There are reserved symbols in XML. The ampersand character becomes & the apostrophe becomes ' the double quote becomes " the less than sign becomes < and the greater than sign becomes &gt. See www.w3.org for a description of the XML standard.

A useful trick for testing the correctness of the XML file is to load it into the Internet Explorer browser (Netscape does not interpret XML files). Alternatively, please validate the XML against the import.xsd schema (found at www.markit.com/import.xsd) in advance – this will also catch missing element errors.

In some cases upon upload, following error might be displayed on the screen:-

> Invalid byte 1 of 1-byte UTF-8 sequence. cp = /usr/java/jdk/lib/tools.jar:/usr/share/tomcat5/bin/bootstrap.jar:/usr/share/tomcat5/bin/commons-logging-api.jar:/usr/share/tomcat5/common/lib/jsk-platform.jar:/usr/share/tomcat5/common/lib/jsk-resources.jar

If you get an error as above, please check for the following in your XML file in order as listed:-

- There are no "smart quotes" (curly double quotes) or trademark symbol (copy pasted from word).
- There is no space at the start of the file.
- The file is encoded in utf-8 or 7 bit ASCII character codes.

In the following examples, the XML file is laid out in a way that makes it easy to read, with extra whitespace in-between the elements and the data. This is purely for display purposes – please remove that whitespace in any files you send. Carriage returns and tabs outside of elements are completely ignored, and can be removed or left in.

The first line of the XML file should look like this:

```
<?xml version="1.0"?>
```

This is standard XML syntax -- it says the file will contain XML and that the XML used is version 1.0. See www.w3.org for more details.

The second line of the XML file should look like this (note that the code is case sensitive):

```
<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.Markit.com/import.xsd">
```

This specifies the schema file used for checking the syntax of the submitted XML file. If Markit upgrade to a new schema file, customers will be able to convert to the new one by amending this line when they are ready. See the error messages section for a description of errors when this line is entered incorrectly.

The <import> section contains header information within the <header> element, followed by the bond and default swap data in the <data> element. Remember that the last line of the XML file will be the closing </import> tag. If this is missing, you will get the error message: 'CRITICAL ERROR - Fatal Error in upload file, line 198, char 1 Message: The input ended before all started tags were ended. Last tag started was 'import'.

The <header> section specifies the system and file ID performing the feed, which for our example looks like this:

```
<header>
   <system> Global      </system>
```

```
    <fileid> 1           </fileid>
    <date>   20-Jun-2002 </date>
</header>
```

The <system> tag contains the name of the system that is feeding the Markit database and was agreed with Markit. The system name is mandatory and case sensitive.

The <fileid> is set to ”1”. In actuality, there may be several feeds from one system, in which case this fileid would be set to the fileid for one of those feeds.

The <date> tag specifies the date for which the data applies. This tag is mandatory and can be in one of the following formats: dd-Mmm-yyyy (e.g. 01-Jan-2006) or yyyymmdd (e.g. 20060101).

Default Swap Mark to Market Feed

Now, we come to the main part of the XML upload file, the <data> element. This element is where default swap and bond data is provided.

Please note that bid and ask prices and spreads can be provided by adding 'bid' or 'ask' to price and spread tags (see example in the enhancements section below). Without 'bid' or 'ask' appended, you will be submitting mid prices.

The default swap feed is enhanced to include an activity and materiality flag, both of which are optionals.

Let’s look at how we would specify a USD denominated TYCO default swap curve.

```
<?xml version="1.0"?>

<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.Markit.com/import.xsd">

<header>
     <system>                Global                      </system>
     <fileid>                1                           </fileid>
     <date>                  20-Jun-2004                 </date>
</header>

<data>

<credswap>
   <entity>                  TYCO </entity>
   <entityname>              Tyco Intl </entityname>
   <tier>                    SNRFOR </tier>
   <baseccy>                 USD </baseccy>
   <docclause>               CR </docclause>
   <credcurve>
      <spread3y>             0.800 </spread3y>
      <spread5y>             0.860 </spread5y>
      <spread10y>            0.900 </spread10y>
      <recovery>             0.5   </recovery>
   </credcurve>
</credswap>
<credswap>
   <entity type=”RED”>       4H98A7 </entity>
   <entityname>              Ford Motor Credit </entityname>
   <tier>                    SNRFOR </tier>
   <baseccy>                 USD </baseccy>
   <docclause>               MR </docclause>
   <credcurve>
      <spread3y>             0.150 </spread3y>
      <spread5y>             0.175 </spread5y>
      <spread10y>            0.210 </spread10y>
      <recovery>             0.5   </recovery>
   </credcurve>
```

```
      <materialitycurve>
         <materiality3y>        Y </materiality3y>
         <materiality5y>        N </materiality5y>
         <materiality10y>      Y </materiality10y>
         <materialityrecovery> N </materialityrecovery>
      </materialitycurve>
      <activitycurve>
         <activity3y>             Y </activity3y>
         <activity5y>             N </activity5y>
         <activity10y>           Y </activity10y>
         <activityrecovery> N </activityrecovery>
      </activitycurve>

   </credswap>

   </data>
```

The following table describes the tags in detail.

Tag	Comments
credswap	Specifies that a default swap curve is being supplied.
entity	Contains the customer's code for identifying the reference entity, and is mandatory.
entityname	Optional and is used to supply a longer description (up to 70 characters) for the reference entity. This is useful if the customer entity codes are not mnemonic. The name provided will appear on the Markit Entity mapping screen.
tier	Mandatory. Must be one of, e.g: SECDOM, SNRFOR, SUBLT2, JRSUBUT2, PREFT1. See Appendix G – Table of Seniority Levels for a full list and explanation of each value.
baseccy	Used to specify the ISO currency code in which the default swap curve is quoted, in this case USD.
docclause	Mandatory. Describes the type of restructuring events that trigger the default swap contract and takes the following values: CR, MR, MM, or XR. See Appendix H – Table of Document Clauses for a full explanation of each value.
credcurve	Contains the details of the credit curve, and the assumed recovery rate for the entity. The following is a list of valid credit curve points: <spread6m>, <spread1y>, <spread2y>, <spread3y>, <spread4y>, <spread5y>, <spread7y>, <spread10y>, <spread15y>, <spread20y>, <spread30y>. A value of 0.0001 represents one basis point. Each spread represents the on-market rate for a transaction with a current start date (T+1) and maturity date equal to the next quarterly roll date following the number of years specified, e.g. on the 5th of May 2004, spread3y means the on-market rate for a trade starting on the 6th of May 2004 and ending on the 20th of June 2007.
recovery	Specifies the recovery rate for this credit and seniority. A value of 0.5 represents 50 percent.
materialitycurve	Material Position means a then-current inventory position of the contributor at the Submission Time which is either of a minimum size of US$20,000,000 or US$10,000 delta (meaning sensitivity to a 1 basis point change in credit spread). Note the minimum notional and delta is subject to change and any coding connected to this field should be amendable.
activitycurve	Active Instruments means, for a given contribution date, instruments flagged by the contributor as those in which they have traded, on that date, a minimum volume of either $20,000,000 notional or $10,000 delta (meaning sensitivity to a 1 basis point change in credit spread). Note the minimum

Tag	Comments
	notional and delta is subject to change and any coding connected to this field should be amendable.

Default Swap Feed Enhancements for Bid/Ask, Quotes, & Trades

The default swap feed is enhanced to handle:

- Bid and Ask default swap curves
- Intra day default swap data with the introduction of the time tag
- Default swap trade and quote data

The following is example XML for the default swap feed handling these items – as shown in red.

```
<?xml version="1.0"?>

<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.markit.com/import.xsd">

<header>
    <system>                    Global          </system>
    <fileid>                    1               </fileid>
    <date>                      20-Jan-2005     </date>
</header>

<data>

<!-- Bid, ask curves provided with bidcurve and askcurve tags-->
<!-- The time tag used to indicate an intra-day curve        -->
<credswap>
    <entity type=”RED”>         748924          </entity>
    <entityname>                AK Corp         </entityname>
    <tier>                      SNRFOR          </tier>
    <baseccy>                   USD             </baseccy>
    <docclause>                 CR              </docclause>
    <time>                      17:00 LDN       </time>
    <bidcurve>
         <spread3y>             0.05            </spread3y>
          <spread5y>            0.05            </spread5y>
          <spread7y>            0.05            </spread7y>
         <recovery>             0.5             </recovery>
    </bidcurve>
    <askcurve>
         <spread3y>             0.05            </spread3y>
         <spread5y>             0.05            </spread5y>
         <spread7y>             0.05            </spread7y>
         <recovery>             0.5             </recovery>
    </askcurve>
</credswap>

<!-- Trade and quote data is supplied within the trade tag    -->
<!-- This example shows a fixed maturity and term to maturity -->
<credswap>
    <entity>                    TYCO            </entity>
    <entityname>                Tyco Intl       </entityname>
    <tier>                      SNRFOR          </tier>
    <baseccy>                   USD             </baseccy>
    <docclause>                 CR              </docclause>
    <quote>
         <time>                 13:20 LDN       </time>
```

```
            <maturity>              2010-03-20      </maturity>
            <spreadbid>             .04             </spreadbid>
            <spreadask>             .042            </spreadask>
            <size>                  10000000        </size>
      </quote>
      <quote>
            <time>                  13:24 LDN       </time>
            <maturity >             2012-03-20      </maturity >
            <spreadbid>             .05             </spreadbid>
            <spreadask>             .05             </spreadask>
            <size>                  10000000        </size>
      </quote>
      <trade>
            <time>                  13:30 LDN       </time>
            <maturity>              2010-03-20      </maturity>
            <spread>                .043            </spread>
            <size>                  -10000000       </size>
      </trade>
      < trade>
            <time>                  14:10 LDN       </time>
            <term>                  5Y              </term>
            <spread>                .05             </spread>
            <size>                  10000000        </size>
      </trade>
</credswap>

</data>

</import>
```

The following table describes in detail the tags for the above XML.

Tag	Comments
time	Used to indicate that the data is intra-day and not end of day mark to market data. The format for the <time> tag is 24 hour HH:MM TMZ, where TMZ is LDN, EUR, NY, TOKYO, SINGAPORE, SYDNEY, or HK. Markit will perform any time zone adjustment to GMT.
bidcurve	For specifying a bid side curve.
askcurve	For specifying an ask side curve.
maturity	For specifying a fixed maturity date for a trade. Provided in yyyy-mm-dd format.
term	For specifying a number of integer years or integer months to maturity.
spread	For specifying the spread at which the trade was transacted. This can appear with the upfront tag. The data is provided as a decimal number, i.e. 0.0001 represents one basis point.
size	The size and direction of the trade, or an indication of the size of the trade. To indicate the trade size, use S for Small (less than $5,000,000 notional); M for Medium ($5,000,000 to $10,000,000 notional); or L for Large ($10,000,000 or greater). Otherwise provide the size and direction of the trade in currency, i.e. -1000000 for a one million dollar default swap sell.

Loan Default Swap Mark to Market Feed

Now, we come to the main part of the XML upload file: the <data> element. The Loan Default Swap Mark to Market Feed is structurally very similar to the Default Swap Mark to Market Feed. This element is where default swap and bond data is provided.

Please note that bid and ask prices and spreads can be provided by adding 'bid' or 'ask' to price and spread tags. Without 'bid' or 'ask' appended, you will be submitting mid prices.

Let's look at how we would specify a USD denominated ALLIED WASTE loan default swap curve.

```
<?xml version="1.0"?>

<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.Markit.com/import.xsd">

<header>
      <system>                Global                        </system>
      <fileid>                1                             </fileid>
      <date>                  20-Jun-2004                   </date>
</header>

<data>

<credswap>
   <entity>                   3250083 </entity>
   <entityname>               ALLIED WASTE INDUSTRIES </entityname>
   <tier>                     LIEN1 </tier>
   <baseccy>                  USD </baseccy>
   <docclause>                XR </docclause>
   <cancellability>           Y </cancellability>
   <credcurve>
      <spread3y>              0.800 </spread3y>
      <spread5y>              0.860 </spread5y>
      <spread10y>             0.900 </spread10y>
      <recovery>              0.7   </recovery>
   </credcurve>
</credswap>
<credswap>
   <entity >                  Ford </entity>
   <entityname>               Ford Motor Company </entityname>
   <tier>                     LIEN1 </tier>
   <baseccy>                  USD </baseccy>
   <docclause>                XR </docclause>
   <cancellability>           Y </cancellability>
   <credcurve>
      <spread3y>              0.090 </spread3y>
      <spread5y>              0.115 </spread5y>
      <spread10y>             0.145 </spread10y>
      <recovery>              0.7   </recovery>
   </credcurve>
</credswap>

</data>
```

The following table describes in detail the tags for the above XML.

Tag	Comments
credswap	Specifies that a default swap curve is being supplied.
entity	Contains the customer's code for identifying the reference entity, and is mandatory.
entityname	Optional and is used to supply a longer description (up to 70 characters) for the reference entity. This is useful if the customer entity codes are not mnemonic. The name provided will appear on the Markit Entity mapping screen.
tier	Mandatory. Must be one of, e.g: LIEN1, LIEN2, LIEN3. See Appendix G – Table of Seniority Levels for a full list and explanation of each value.
baseccy	Used to specify the ISO currency code in which the default swap curve is quoted, in this case USD.
docclause	Mandatory. Describes the type of restructuring events that trigger the default swap contract and takes the following values: MR, MM, or XR. See Appendix H – Table of Document Clauses for a full explanation of each value.
cancellability	Refers to whether the contract is the cancellable European version (protection terminates upon refinancing) represented by "Y" or the non-cancellable North American version, represented by a "N" or the Bullet cancellable version, represented by a "B".
credcurve	Contains the details of the credit curve, and the assumed recovery rate for the entity. The following is a list of valid credit curve points: <spread6m>, <spread1y>, <spread2y>, <spread3y>, <spread4y>, <spread5y>, <spread7y>, <spread10y>, <spread15y>, <spread20y>, <spread30y>. A value of 0.0001 represents one basis point. Each spread represents the on-market rate for a transaction with a current start date (T+1) and maturity date equal to the next quarterly roll date following the number of years specified, e.g. on the 5th of May 2004, spread3y means the on-market rate for a trade starting on the 6th of May 2004 and ending on the 20th of June 2007.
recovery	Specifies the recovery rate for this credit and seniority. A value of 0.7 represents 70 percent.

Loan Default Swap Feed Enhancements for Bid/Ask

The default swap feed is enhanced to handle:

- Bid and Ask default swap curves
- Intra day default swap data with the introduction of the time tag

The following is example XML for the default swap feed handling these items—as shown in red.

```
<?xml version="1.0"?>

<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.markit.com/import.xsd">

<header>
    <system>                    Global          </system>
    <fileid>                    1               </fileid>
    <date>                      20-Jan-2005     </date>
</header>

<data>

<!-- Bid, ask curves provided with bidcurve and askcurve tags-->
<!-- Curves contain an upfront element on the three year     -->
```

```
<!-- The time tag used to indicate an intra-day curve        -->
<credswap>
    <entity>                    3250083 </entity>
    <entityname>                ALLIED WASTE INDUSTRIES </entityname>
    <tier>                      LIEN1 </tier>
    <baseccy>                   USD </baseccy>
    <docclause>                 XR </docclause>
    <cancellability>             Y </cancellability>
    <time>                        17:00 LDN          </time>
    <bidcurve>
          <spread3y>              0.22               </spread3y>
          <spread5y>              0.19               </spread5y>
          <spread7y>              0.15               </spread7y>
          <recovery>              0.7                </recovery>
    </bidcurve>
    <askcurve>
          <spread3y>              0.27               </spread3y>
          <spread5y>              0.14               </spread5y>
          <spread7y>              0.20               </spread7y>
          <recovery>              0.7                </recovery>
    </askcurve>
</credswap>

<!-- Trade and quote data is supplied within the trade tag    -->
<!-- This example shows a fixed maturity and term to maturity -->
<credswap>
    <entity >                     Ford </entity>
    <entityname>                  Ford Motor Company </entityname>
    <tier>                        LIEN1 </tier>
    <baseccy>                     USD </baseccy>
    <docclause>                   XR </docclause>
    <cancellability>               Y </cancellability>
    <quote>
          <time>                  13:20 LDN          </time>
          <maturity>              2010-03-20         </maturity>
          <spreadbid>             .05                </spreadbid>
          <spreadask>             .06                </spreadask>
    </quote>
    <quote>
          <time>                  13:20 LDN          </time>
          <maturity >             2010-03-20         </maturity >
          <spreadbid>             .05                </spreadbid>
          <spreadask>             .06                </spreadask>
    </quote>

</credswap>

</data>

</import>
```

The following table describes in detail the tags for the above XML.

Tag	Comments
time	Used to indicate that the data is intra-day and not end of day mark to market data.

Tag	Comments
	The format for the <time> tag is 24 hour HH:MM TMZ, where TMZ is LDN, EUR, NY, TOKYO, SINGAPORE, SYDNEY, or HK. Markit will perform any time zone adjustment to GMT.
bidcurve	For specifying a bid side curve.
askcurve	For specifying an ask side curve.
upfront	For specifying the upfront payment in a high yield credit. The data is provided as a decimal number, i.e. 0.0001 represents one basis point.
maturity	For specifying a fixed maturity date for a trade. Provided in yyyy-mm-dd format.
term	For specifying a number of integer years or integer months to maturity.
upfront	For specifying any upfront payment in the trade expressed as a percentage of notional. This can appear with the spread tag. The data is provided as a decimal number, i.e. 0.0001 represents one basis point.
spread	For specifying the spread at which the trade was transacted. This can appear with the upfront tag. The data is provided as a decimal number, i.e. 0.0001 represents one basis point.
size	The size and direction of the trade, or an indication of the size of the trade. To indicate the trade size, use S for Small (less than $5,000,000 notional); M for Medium ($5,000,000 to $10,000,000 notional); or L for Large ($10,000,000 or greater). Otherwise provide the size and direction of the trade in currency, i.e. -1000000 for a one million dollar default swap sell.

Cash Bond Feed

We now come to the bond. A bond is denoted with the <bond> tag. There are several examples: the first example is delivered with a bond price (bid/offer or mid examples provided); the second example with a local currency asset swap spread; and the remaining examples describe how to submit data expressed as a spread to a benchmark yield.

```
<?xml version="1.0"?>

<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.Markit.com/import.xsd">

<header>
     <system>                    Global                  </system>
     <fileid>                    1                       </fileid>
     <date>                      20-Jun-2004             </date>
</header>

<data>

<!—- Data expressed as a Bid/Offer cash price -->
<bond>
    <bondname>           RATP 5.75 25-OCT-10          </bondname>
    <bondid type="isin"> FR0000482721                 </bondid>
    <book>               Finvision                    </book>
    <snaptime>           16:30 LDN                    </snaptime>
    <pricebid>           1.02153762                   </pricebid>
    <priceask>           1.02291526                   </priceask>
    <position>           L                            </position>
```

```
</bond>


<!-- Data expressed as a mid cash price -->
<bond>
   <bondname>             GLOPAR 10.625 05Dec08 </bondname>
   <bondid type="cusip">  37957TAD </bondid>
   <book>                 FYEUR </book>
   <bondprice>
      <time>              17:00 </time>
      <price type="cash"> 0.875 </price>
   </bondprice>
   <position> L </position>
</bond>

<!-- Asset swap spread -->
<bond>
   <bondname>             PACLIF 5.5 28May09 </bondname>
   <bondid type="cusip">  TT9379837</bondid>
   <book>                 AEUE </book>
   <bondprice>
      <time>                   17:00 GMT</time>
      <spread type="localpar"> 0.021 </spread>
     <position>                  F </position>
 </bondprice>
</bond>

<!-- Spread over the interpolated benchmark curve for this -->
<!-- bond's currency  (e.g. Treasury for USD Corps)        -->
<bond>
     <bondname>                       NW 3.5 Aug15       </bondname>
     <bondid type="isin">             XS0212575462       </bondid>
     <book>                           NY-PROP            </book>
     <snaptime>                       16:30 NY           </snaptime>
     <interpbenchspreadbid>           0.002254           </interpbenchspreadbid>
     <interpbenchspreadask>           0.002254           </interpbenchspreadask>
</bond>

<!-- Spread over a specific benchmark bond -->
<bond>
     <bondname>                       IBP 6.0 Mar20      </bondname>
     <bondid type="isin">             XS0213104630       </bondid>
     <book>                           NY-PROP            </book>
     <snaptime>                       16:30 NY           </snaptime>
     <benchspreadbid>                 0.002254           </benchspreadbid>
     <benchid type="isin">            XS0213101230       </benchid>
</bond>

<!-- Mid spread over a specific benchmark with yield provided -->
<bond>
     <bondname>                       G 5.5 Apr15        </bondname>
     <bondid type="isin">             XS723423944        </bondid>
     <book>                           NY-PROP            </book>
     <snaptime>                       16:30 NY           </snaptime>
     <benchspread>                    0.002254           </benchspread>
     <benchyield>                     0.03               </benchyield>
</bond>

</data>
```

The following table describes the tags in detail.

Tag	Comments
bond	Specifies that the following data applies to a Bond.
bondname	Optional. Used to provide better error messages in the event that there is a problem during upload (for example, if the bond details are not in the Markit database).
bondid	Mandatory. Used to uniquely identify the bond. The particular type of ID is specified with the 'type' attribute, which can be one of the following: "isin"; "cusip"; "common"; "fonds"; "sedol"; "sicovam"; "svm"; "valoren"; "wpk".
book	Mandatory, and describes the trading book or desk from which the price was sourced.
bondprice	Contains the information required for specifying a bond price, and the time at which the bond price was recorded.
time	24 hour HH:MM TMZ, where TMZ is LDN, EUR, NY, TOKYO, SINGAPORE, SYDNEY, or HK. Markit performs any time zone adjustment to GMT.
price	The bond's clean price in decimal, i.e. 0.9 for 90 percent. Negatives and zero are not permitted.
pricebid	The bond's clean Bid price in decimal, i.e. 0.9 for 90 percent. Negatives and zero are not permitted.
priceask	The bond's clean Ask price in decimal, i.e. 0.9 for 90 percent. Negatives and zero are not permitted.
spread	Describes the price as a spread. The type of spread is described with an attribute: "localpar" for an asset swap spread denominated in the currency of the bond, or "usdpar" for an asset swap spread that has been adjusted to USD with a conversion factor and the basis swap.
position	Optional. The position as indicated by the letter 'L' for a large position ($5,000,000 or more); the letter 'S' for a small position; or the letter 'F' when, although there is no position behind the price, the price is a firm one.

When providing data expressed as a spread to a benchmark bond yield (the convention used for US Corporates, which are expressed as a spread to Treasury bonds), there are three options:

1. A spread over a specific benchmark bond which is provided in the <benchspread> tag (add 'bid' or 'ask' to this tag name for bid and ask spreads) along with the ID of the bond with the <benchid> tag. The bond's yield is extracted from the Markit database which records changes in government bond yields in real-time.
2. A spread over an interpolated benchmark curve, in which case the spread of the corporate bond is provided in the <interpbenchspread> tag and Markit adds this data to their internally held benchmark government bond curve which is stored in real-time in the database.
3. A spread over a bond where the bond's yield is supplied. In this case, the <benchspread> is used to provide the corporate bond's spread to the Treasury and the <benchyield> tag is used to provide the yield of the benchmark. Note that you can add 'bid' or 'ask' to the tag names for specifying bid or ask quotes.

When publishing prices as a spread to a Treasury bond, the system uses the following algorithm for determining the Treasury against which to spread, unless over half of contributors have chosen another benchmark.

Time to maturity	Chosen Benchmark Bond
> 15 years	Spread to the 30 year on the run benchmark

Time to maturity	Chosen Benchmark Bond
9 to 15 years	Spread to the 10 year on the run benchmark
4 to 8 years	Spread to the 5 year on the run benchmark
1 to 4 years	Spread to the 1 year on the run benchmark

Convertible Bond Feed

This section describes the construction of a convertible bond feed. Here is a sample of the XML:

```
<?xml version="1.0"?>

<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.Markit.com/import.xsd">

<header>
   <system>                    Global                    </system>
   <fileid>                    1                         </fileid>
   <date>                      20-Jun-2004               </date>
</header>

<data>

<convert>
   <bondname>                  GLOPAR 5% 05Dec08         </bondname>
   <bondid type="cusip">       37967TAD                  </bondid>
   <book>                      FYEUR                     </book>
   <snaptime>                  17:00 LDN                 </snaptime>
   <pricebid type="par">       0.87                      </pricebid>
   <priceask type="par">       0.88                      </priceask>
   <equitymid>                 10.5                      </equitymid>
   <equityimpliedvol>          0.3                       </equityimpliedvol>
   <equitydelta type="unit">   0.02                      </equitydelta>
   <creditspread>              0.031                     </creditspread>
   <fx>                        0.0343                    </fx>
   <position>                  L                         </position>
  <conversionratio>            60                        </ conversionratio >
</convert>

</data>
```

The tags in the example XML above are described in the following table.

XML Tag	Type and Validation	Notes
bondname	Optional. Char(100)	A description of the bond which is only used when reporting an error during the automated upload process.
bondid	Mandatory. Used to uniquely identify the bond. The particular type of ID is specified with the 'type' attribute, which can be one of the following: "isin"; "cusip"; "common"; "fonds"; "sedol"; "sicovam"; "svm"; "valoren"; "wpk".	The bond ID.
book	Char(10) Mandatory	The trading book from which this bond price was taken.

XML Tag	Type and Validation	Notes
snaptime	Mandatory. HH:MM ZONE	The time at which the convertible price and other market data was snapped. ZONE can be one of: LDN, EUR, NY, TOKYO, SINGAPORE, or HK. Markit then adjusts this time to GMT inclusive of daylight savings changes.
pricebid	Mandatory. 1 for 100%	The closing bid price for the convertible bond. Note bid and offer must be provided. This has a type attribute that specifies how the price is being quoted, either type="par" to indicate that the price is expressed as a percent of par, or type="unit" to indicate that the price is expressed in terms of a unit of equity.
priceask	Mandatory. 1 for 100%	The closing ask price for the convertible bond. Note bid and offer must be provided.
equitymid	Mandatory	The equity mid price at the time the convertible bond price was provided. This amount is expressed in local currency. For sterling, it is expressed in pounds and pence.
equityimpliedvol	0.2 for 20%	The equity implied volatility.
equitydelta	0.1 for 10%	The sensitivity of the convertible bond to changes in the underlying equity price. This value has a type attribute to specify how the delta is being quoted, which can take the following values: **"unit"** to indicate that the delta is expressed as change in bond price / one unit (Dollar, Euro, Sterling, etc). This is the default quoting type if the type is not specified. **"percent"** to indicate that the delta is expressed as a percentage change in price of the bond / percentage change in equity price. **"parity"** to indicate that the delta is expressed as the price change in the bond divided by (change in stock price * conversion ratio)
creditspread	0.0001 for 1bp	The pure credit spread for the convertible bond's cashflows. From this spread, the bond's fixed income value (known as the bond floor) can be calculated by performing a 'risky discount'.
fx	Number	The FX assumption used when pricing bonds that have the fixed income component in a different currency to the equity. Note this number must be

XML Tag	Type and Validation	Notes
		provided as an FX multiplier.
position	'L' for a Large position over $5 million. 'S' for a Small position. 'F' for a Firm price. Otherwise the actual position can be supplied. Optional	The position underlying the price. This position is used as information when publishing prices back to the website.
conversionratio	Number	The current conversion ratio from the convertible bond into the underlying equity, taking into account any adjustments since issuance.

Credit Index and Index Tranche XML Feed

This section describes the construction of a credit index and index tranche feed. Here is a sample of the XML:

```
<?xml version="1.0"?>

<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.Markit.com/import.xsd">

<header>
       <system>   Global         </system>
       <fileid>   1              </fileid>
       <date>     20-Jan-2005    </date>
</header>

<data>

<!--Credit Index data is provided within the credindex tag-->
<credindex>
       <red>      123456789      </red>
       <name>     CDX NA IG      </name>
       <term>     5Y             </term>

       <!-- An intra-day quote expressed as a running spread -->
       <!-- and an upfront                                   -->
       <quote>
              <time>             14:08 NY   </time>
              <upfrontbid>       0.45       </upfrontbid>
              <upfrontask>       0.50       </upfrontask>
              <spread>           0.05       </spread>
       </quote>
       <!--A trade expressed as a spread-->
       <trade>
              <time>             14:08 NY   </time>
              <spread>           0.067      </spread>
              <size>             10000000   </size>
       </trade>

       <!--A closing mark expressed as a price or upfront-->
       <close>
              <time>             14:08 NY   </time>
              <upfront>          0.30       </upfront>
              <spread>           0.067      </spread>
       </close>
```

```
</credindex>
<!--Credit Index tranche data -->
<credtranche>
       <red>      123456789      </red>
       <name>     NAIG Equity    </name>
       <attach>   0              </attach>
       <detach>   0.03           </detach>
       <term>     5Y             </term>

       <!-- An closing mark expressed as a running spread -->
       <!-- and an upfront                                -->
       <close>
              <time>            17:00 NY   </time>
              <upfront>         0.46       </upfront>
              <spreadbid>       0.055      </spreadbid>
              <spreadask>       0.07       </spreadask>
              <correlationbid>  0.27       </correlationbid>
              <correlationask>  0.29       </correlationask>
              <delta>5</delta>
              <!--The price of the index is also provided-->
              <credindex>
                     <time>       17:00 NY  </time>
                     <upfront>    0.4       </upfront>
                     <spread>     0.05      </spread>
              <credindex>
       </close>

     <!-- An intra-day quote expressed as a running spread -->
       <!-- and an upfront                                   -->
       <quote>
              <time>            14:08 NY   </time>
              <upfront>         0.45       </upfront>
              <spreadbid>       0.05       </spreadbid>
              <spreadask>       0.07       </spreadask>
              <correlationbid>  0.275      </correlationbid>
              <correlationask>  0.29       </correlationask>
              <delta>1</delta>
              <!--The price of the index is also provided-->
              <credindex>
                     <time>       14:08 NY  </time>
                     <upfront>    0.4       </upfront>
                     <spread>     0.05      </spread>
              <credindex>
       </quote>

       <!--A trade expressed as a spread-->
       <trade>
              <time>            14:08 NY   </time>
              <spread>          0.067      </spread>
              <delta>1</delta>
              <credindex>
                     <time>       14:08 NY  </time>
                     <upfront>    0.4       </upfront>
                     <spread>     0.05      </spread>
              <credindex>
              <size>  10000000       </size>
       </trade>

       <!--A trade expressed as a price or upfront-->
```

```
        <trade>
                <time>          14:08 NY   </time>
                <upfront>       0.30       </upfront>
                <spread>        0.067      </spread>
                <delta>1</delta>
                <credindex>
                        <time>      14:08 NY  </time>
                        <upfront>   0.4       </upfront>
                        <spread>    0.05      </spread>
                <credindex>
                <size>          10000000   </size>
        </trade>
</credtranche>

</data>

</import>
```

The following table provides a description of the tags in the above XML example.

Tag	Comments
credindex	Specifies that a credit index is being supplied.
red	The RED ID used to identify the credit index.
name	The name of the credit index instrument. This name is used to improve the readability of an error message should an error occur and is optional.
spread	The spread of the credit index.
term	The term to maturity of the index – e.g. '5Y' for a 5-year index.
time	Indicates that the data is intra-day and not end of day mark to market data. The format for the <time> tag is 24 hour HH:MM TMZ, where TMZ is LDN, EUR, NY, TOKYO, SINGAPORE, SYDNEY or HK. Markit performs any time zone adjustment to GMT.
upfrontbid	The upfront bid payment. The data is provided as a decimal number, i.e. 0.0001 represents one basis point.
upfrontask	The upfront ask payment. The data is provided as a decimal number, i.e. 0.0001 represents one basis point.
spreadbid	The bid spread. The data is provided as a decimal number, i.e. 0.0001 represents one basis point.
spreadask	The ask spread. The data is provided as a decimal number, i.e. 0.0001 represents one basis point.
delta	The sensitivity of the tranche to the underlying index, i.e. number of BP change in the tranche spread for a 1bp change in the index spread.
size	The size and direction of the trade or an indication of the size of the trade. To indicate the trade size use S for Small (less than $5,000,000 notional); M for Medium ($5,000,000 to $10,000,000 notional); or L for Large ($10,000,000 or greater). Otherwise provide the size and direction of the trade in currency, i.e. -1000000 for a one million dollar default swap sell.

Tag	Comments
attach	The index tranche attachment point.
detach	The index tranche detachment point.
correlationbid	The implied correlation for the bid price / spread of an index tranche.
correlationask	The implied correlation for the ask price / spread of an index tranche.

Asset-Backed Securities (ABS) Feed - Import

The ABS feed provides for an asset-backed security's price, spread, and average life to be provided to Markit. Instrument spreads, expressed as Discount Margins or Asset Swap spreads, can be relative to a treasury (as with UK asset-backed securities) or a swap curve. Where the instrument spread is relative to a specific treasury, this information should be recorded, too. In the following sample code, the first <abs> shows an instrument spread to swaps, while the second <abs> shows an instrument spread to treasury.

```
<?xml version="1.0"?>

<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.markit.com/import.xsd">

<header>
       <system>       Global  </system>
       <fileid>       1       </fileid>
       <date>  20-Jan-2005    </date>
</header>

<data>

<abs>
     <name>                          CONSECO 98/2 Class C          </name>
     <id type="isin">                8765437967TA                  </id>
     <book>                          FYEUR                         </book>
     <snaptime>                      17:00 LDN                     </snaptime>
     <pricebid>                      0.87                          </pricebid>
     <priceask>                      0.88                          </priceask>
     <benchspreadbid>                0.031                         </benchspreadbid>
     <benchspreadask>                0.028                         </benchspreadask>
     <avlife>                        5.3                           </avlife>
     <prepay type="cpr">             0.09                          </prepay>
     <position>                      L                             </position>
</abs>

<abs>
     <name>                          CONSECO 98/2 Class C          </name>
     <id type="isin">                8765437967TA                  </id>
     <book>                          FYEUR                         </book>
     <snaptime>                      17:00 LDN                     </snaptime>
     <price>                         0.87                          </price>
     <benchspread>                   0.031                         </benchspread>
     <benchspreadyield>              0.034                         </benchspreadyield>
     <benchid type="isin">           XS0213104630                  </benchid>
     <avlife>                        5.3                           </avlife>
     <prepay type="cpr">             0.09                          </prepay>
     <position>                      L                             </position>
</abs>

</data>

</import>
```

The tags in the above example XML are described in the following table.

XML Tag	Type and Validation	Notes
name	Optional. Char(100)	A description of the instrument.
id	Mandatory. Used to uniquely identify the bond. The particular type of ID is specified with the 'type' attribute, which can be one of the following: "isin"; "cusip"; "common"; "fonds"; "sedol"; "sicovam"; "svm"; "valoren"; "wpk".	The bond ID.
book	Mandatory. Char(10)	The trading book from which this instrument price was taken.
snaptime	HH:MM ZONE Mandatory	The time at which the instrument price and other market data was snapped. ZONE can be one of: LDN, EUR, NY, TOKYO, SINGAPORE, HK. Markit then adjusts the time to GMT taking into account daylight savings.
pricebid price priceask	Mandatory. 1 for 100%	The closing price for the instrument. Note that bid, offer, or mid can be provided (although the user must have filled in Bid/Ask Adjustment form for this to work with anything other than both bid and ask).
benchspreadbid benchspread benchspreadask	Mandatory. 1 for 100%	The closing spread for the instrument. Note that bid, offer, or mid can be provided (although the user must have filled in Bid/Ask Adjustment form for this to work with anything other than both bid and ask).
benchid	Mandatory. Attribute 'type' takes following values: isin"; "cusip"; "common"; "fonds"; "sedol"; "sicovam"; "svm"; "valoren"; "wpk".	The ID of the relevant treasury bond.
benchyield	Optional. 1 for 100%	The yield of the relevant treasury bond recorded at the above snaptime.
avlife	Mandatory. Float	The weighted average life of the instrument.
prepay	Optional. 1 for 100% Attribute 'type' takes following values: cpr, psa, smm, pps, mpr, cpy	Prepayment speed and type of measure.
position	Optional. L/M/S or a +ve or –ve value	L/M/S characterizes a large, medium, or small position. Also a notional value can be provided (where +ve notional represents a long position).

Asset-Backed Securities (ABS) Feed - Export

The following parameters are for retrieving a download of ABS data.

Parameter (case sensitive)	Values (case insensitive)	Description
report	Composites	The composites report is a download of the composite pricing you are permissioned to view.
	Contributions	The contributions report is a download of the individual price contributions that you are permissioned to view
	Dataquality	As a contributor, you also receive a report informing you which of your prices passed, which failed and which test they failed.
Type	abs	Specifies that this report is an ABS report.
Version	5	The latest available version of this report.

ABS Bid/Ask Spread Adjustments

To facilitate dealers in providing both bid and ask prices on instruments where they typically only provide a mid or a bid price, Markit has allowed contributors to specify a set of adjustments to be applied to instruments based on their ratings and sectors. These adjustments can be multiplicative or additive, and a different parameter can be provided for each combination of price, spread, sector, and rating. Adjusted values are calculated from the mid point according to the following formulas. This method makes the process invertible.

Multiplicative (negative mid point):

$$b = m\left(1 + \frac{f(r,s)}{2}\right)$$

$$a = m\left(1 - \frac{f(r,s)}{2}\right)$$

Multiplicative (positive mid point):

$$b = m\left(1 - \frac{f(r,s)}{2}\right)$$

$$a = m\left(1 + \frac{f(r,s)}{2}\right)$$

Additive (all cases):

$$a = m + \frac{f(r,s)}{2}$$

$$b = m - \frac{f(r,s)}{2}$$

Where:
f is the adjustment factor (given a rating, r, and a sector, s)
B is the bid price/spread
A is the ask price/spread
M is the mid price/spread

The template is available upon request and provides for adjustments in the following sectors and ratings:

ABS Ratings

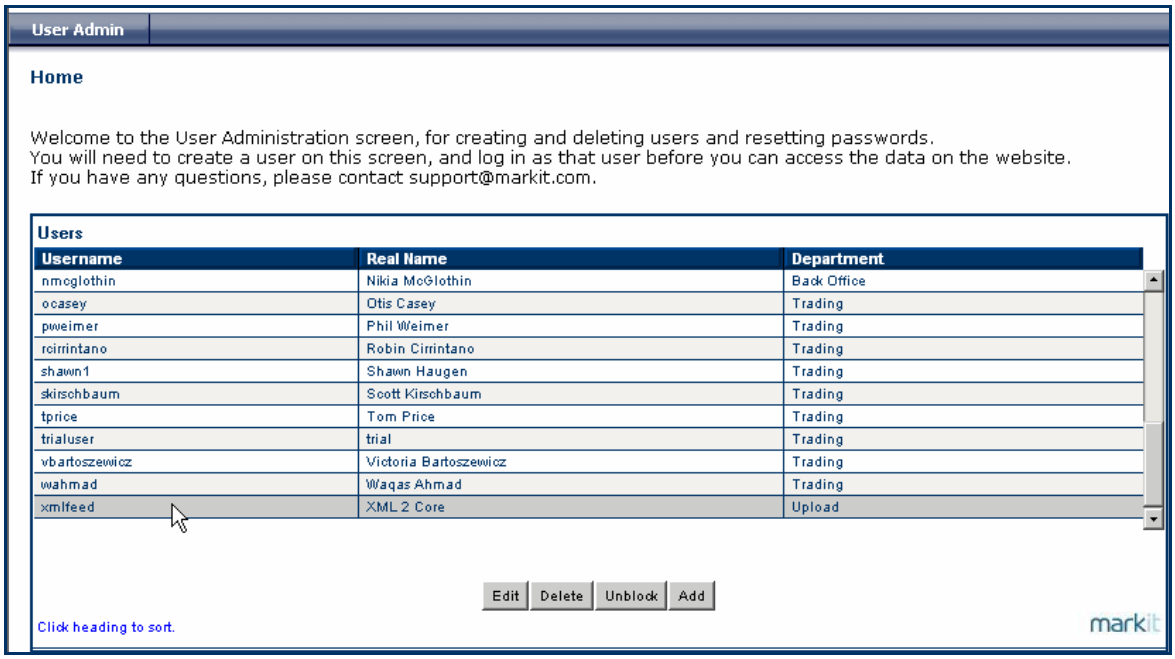
- AAA
- AA
- A
- BBB
- BB
- B
- Other

ABS Sectors

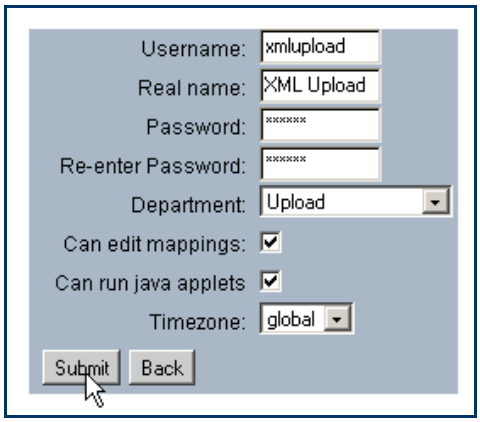
- Auto Lease
- Auto Loan
- CDO
- CMBS
- Credit Card
- Dealer Floorplan
- Equip Lease
- Franchise Loan
- Home Equity
- Insurance Premium Loan
- Manufactured Housing
- Marine Loan
- Motorcycle Loan
- Other
- Other Consumer Loan
- Receivables
- Recreational Vehicle
- RMBS
- Small Business Loan
- Student Loan
- Time Share Loan
- Trade Receivable
- Truck Loan
- Whole Business Securitization

Creating a User with the Ability to Upload

To create a user who is able to upload an XML file to Markit, you must first log into www.Markit.com as the administrator using the login and password provided by Markit. After agreeing to our Terms and Conditions, a screen similar to the example below displays:

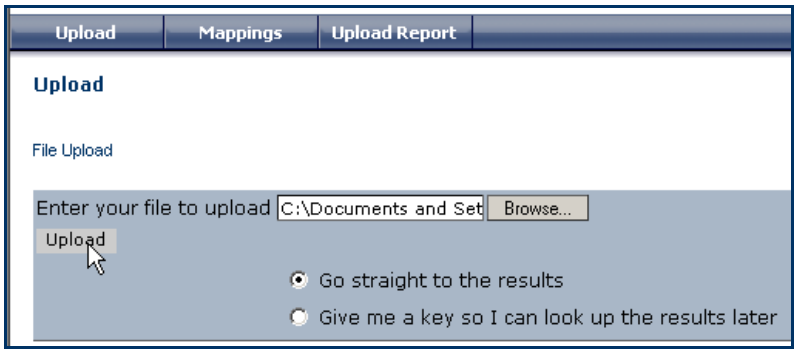


Click the **Add** button to create a new user. To create a user that is able to perform an XML upload, set the user's department to 'Upload'. The password for Upload users does not expire. In addition, we allow this user to maintain and edit the credit default swap entity mappings if you select the 'Can edit mappings' check box. In actuality, this function is typically assigned to another user, but for the purposes of this example, we provide both abilities to the same user. Below is an example of the Create User screen. Click the **Submit** button to create this user.



Uploading an XML File via the Web

Log off as the administrator (click 'Log off' on the top right of the screen) and log in as the user created in the previous example. The Upload screen displays, as in the following example screen.



This screen is used to manually transmit an XML file to the Markit.com website. This process can be automated later, but during development, this screen is useful for quickly validating your XML file.

Type the name of the XML file you want to upload in the 'Enter your file to upload' input box, or click the **Browse** button and select the file.

The system processes your information immediately when you click the **Upload** button. You can either view the results of the upload immediately by selecting the 'Go straight to the results' option (useful when developing), or you can retrieve the results asynchronously with a unique key provided by the system if you choose the 'Give me a key so I can look up the results later' option. (Use this option when automating large feeds, since the network connection may time out before the upload has finished. See the 'asynch=Y' option in the section Automating XML File Transmission to Markit.

For this example, we choose 'Go straight to the results'.

Having specified the XML filename and transmitted it to the Markit database with the **Upload** button, the data is validated and inserted into the Markit database. This process may take a few minutes.

Any errors that occurred during the upload are displayed on the screen as in the following example:

```
File passed XML validation
Started parse on THU SEP 19 18:59:49 2002
Data is being uploaded for 19-SEP-2002
ERROR:ENTITY - Entity AOL has a match within a hierarchy near line 28. The curve will not be inserted until the mapping is confirmed
ERROR:ENTITY - Unable to find match for entity TYCO near line 84.
ERROR:ENTITY - Unable to find match for entity BARIC near line 176.
ERROR:IMPORT - ORA-20000: BondMtm(): No such instrument 'RAIT-P 7.375 18Nov22', CUSIP='TT3354201' for book 'Demo' near line 95.
ERROR:IMPORT - ORA-20000: BondMtm(): No such instrument '-', CUSIP='3133MJH60' for book 'Demo' near line 105.

Completed parse on THU SEP 19 19:00:29 2002

STATISTICS
==========
No. of data elements in file:                2570
No. of valid data elements in file:          2318


No. of bond prices inserted into database:   2316
No. of bond prices rejected by database:     2

No. of invalid bonds supplied:               0
No. of bonds not found in Mark-it database:  2

No. of curves inserted into database:        2048
No. of curves rejected by database:          0
No. of recoveries inserted into database:    2048
No. of recoveries rejected by database:      0
No. of converts inserted into database:      0
No. of converts rejected by database:        0
```

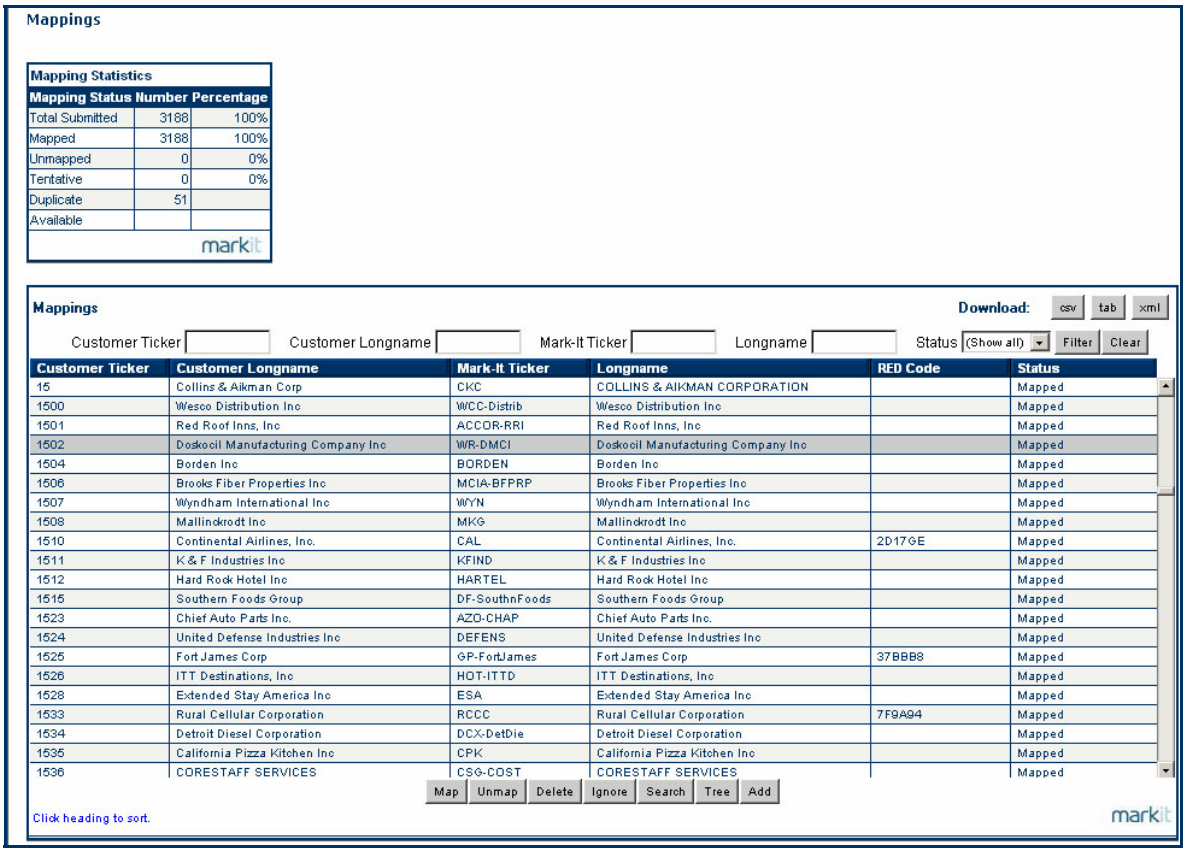
In the above example, we can see from the status report that a number of default swap entity codes require mapping to their equivalent Markit codes (AOL, TYCO and BARIC). Errors of this kind are resolved in the entity mapping screen which we will look at next.

In addition, two bond prices were transmitted, but for bonds that were not found in the Markit database. The error message for a missing bond is improved if the XML file includes a descriptive bond name with the <bondname> tag. So, for example, in the previous screen, the bond with Cusip 3133MJH60 was not provided with a bond description, and so a bond name does not appear in the error.

At the end of the status report is a summary section outlining overall statistics for the file transfer. Bonds not found in the Markit database are retrieved from our static bond data provider.

Mapping Customer Entity Codes to Markit Entity Codes

Errors with default swap entity codes are resolved with the Markit Entity mapping screen, which is accessed by choosing the Mappings option at the top of the screen (access to this screen is granted using the Can Edit Mappings? feature on the User Maintenance screen). This screen is populated with entities that were reported during the upload process. So, continuing with our example, the screen looks like this:



Mappings

Mapping Statistics

Mapping Status	Number	Percentage
Total Submitted	3188	100%
Mapped	3188	100%
Unmapped	0	0%
Tentative	0	0%
Duplicate	51	
Available		

markit

Mappings — Download: csv tab xml

Customer Ticker | Customer Longname | Mark-It Ticker | Longname | Status (Show all) | Filter | Clear

Customer Ticker	Customer Longname	Mark-It Ticker	Longname	RED Code	Status
15	Collins & Aikman Corp	CKC	COLLINS & AIKMAN CORPORATION		Mapped
1500	Wesco Distribution Inc	WCC-Distrib	Wesco Distribution Inc		Mapped
1501	Red Roof Inns, Inc	ACCOR-RRI	Red Roof Inns, Inc		Mapped
1502	Doskocil Manufacturing Company Inc	WR-DMCI	Doskocil Manufacturing Company Inc		Mapped
1504	Borden Inc	BORDEN	Borden Inc		Mapped
1506	Brooks Fiber Properties Inc	MCIA-BFPRP	Brooks Fiber Properties Inc		Mapped
1507	Wyndham International Inc	WYN	Wyndham International Inc		Mapped
1508	Mallinckrodt Inc	MKG	Mallinckrodt Inc		Mapped
1510	Continental Airlines, Inc.	CAL	Continental Airlines, Inc.	2D17GE	Mapped
1511	K & F Industries Inc	KFIND	K & F Industries Inc		Mapped
1512	Hard Rock Hotel Inc	HARTEL	Hard Rock Hotel Inc		Mapped
1515	Southern Foods Group	DF-SouthnFoods	Southern Foods Group		Mapped
1523	Chief Auto Parts Inc.	AZO-CHAP	Chief Auto Parts Inc.		Mapped
1524	United Defense Industries Inc	DEFENS	United Defense Industries Inc		Mapped
1525	Fort James Corp	GP-FortJames	Fort James Corp	37BBB8	Mapped
1526	ITT Destinations, Inc	HOT-ITTD	ITT Destinations, Inc		Mapped
1528	Extended Stay America Inc	ESA	Extended Stay America Inc		Mapped
1533	Rural Cellular Corporation	RCCC	Rural Cellular Corporation	7F9A94	Mapped
1534	Detroit Diesel Corporation	DCX-DetDie	Detroit Diesel Corporation		Mapped
1535	California Pizza Kitchen Inc	CPK	California Pizza Kitchen Inc		Mapped
1536	CORESTAFF SERVICES	CSG-COST	CORESTAFF SERVICES		Mapped

Map | Unmap | Delete | Ignore | Search | Tree | Add

Click heading to sort.

markit

➲ **NOTE**: See the Mappings section in the Markit.com User Guide for more information about using this feature. The Markit.com User Guide is available on the Markit.com website on the Help page (click **Help** in the upper right corner of the screen to access this page).

The upload error report referred to the customer codes ‘AOL’ and ‘BARIC’ among others.

The customer code ‘AOL’ was found in the Markit database, but since it was found to be within a company hierarchy in the Markit database, the Mapping status is set to Assumed. It is important that the correct entity is mapped within the hierarchy. Click the **Tree** button to view the corporate hierarchy. Click **Search** to find another ticker to the one that was Assumed or click **Map** to quickly map the Assumed Markit ticker to the Customer Ticker.

The customer codes ‘BC’ and ‘BMY’ were found in the Markit database and since they are not part of a corporate hierarchy and there is no ambiguity about which company this may be, the Mapping status is set to Matched. Click **Map** to quickly map the Matched Markit ticker to the Customer Ticker.

The customer code ‘BARIC’ was not found in the Markit database, so it is set to a status of Not Found and its mapping must be manually resolved. To manually resolve a mapping, click **Search** and search the Markit database, select the matching Markit ticker, and then click **Map** on the Search screen.

Please note that credit curves are recorded in the database only for Mapped tickers.

The Customer Longname column displays the customer’s long entity names, which are supplied with the entityname tag in the XML file.

The Markit Ticker column displays the Markit entity code that corresponds to the customer’s ticker. Where this field is blank, the customer must map it using the **Search** button.

The Longname column displays either the very accurate long name from the RED database (if the customer is a RED subscriber) or the Markit abbreviated name.

The **Filter** and **Clear** buttons are used to control the filtering of data using the various columns. Click the **Clear** button to clear the filter.

The Status column displays the status of the mapping between the customer entity code and the Markit entity code. It is only when the status reads 'Mapped' that default swap data feeds into the Markit database. The following table provides a description of each status code:

Status	Meaning
Mapped	The customer entity code is mapped to a corresponding Markit entity code and default swap data that references this entity feeds through to the Markit database. The customer clicked the **Map** button for a selected code with a Markit ticker entered.
Unmapped	The customer clicked the **Unmap** button for a selected code to unmap a previously Mapped code.
Matched	A matching Markit code was found and it is not within a hierarchy.
Assumed	A matching Markit code was found but it is within a hierarchy. Special attention must be given to selecting the correct entity within the hierarchy.
Not Found	No Markit entity was found with a code that matches the customer code.
Ignore	The customer clicked the **Ignore** button for a selected code to set the status of a code to Ignore. The default swap curve for this entity is ignored during the upload process. This status is useful for credit curves that represent basket trades, etc.

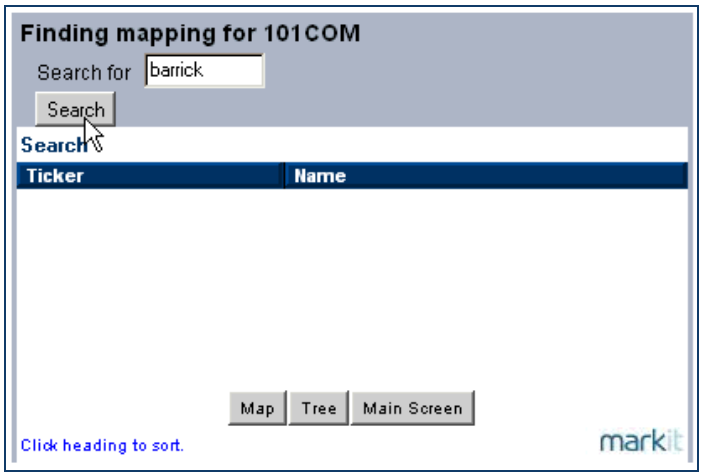
The Markit database is updated with corporate actions and this information is presented in the Status column using the following three codes:

Status	Meaning
Merged	The entity was part of a merger, and the customer may assign a new Markit code.
De-Merged	The entity was part of a de-merger, and the customer may assign a new Markit code.
Bankrupt	The entity is bankrupt, and data is no longer accepted by Markit for this code.

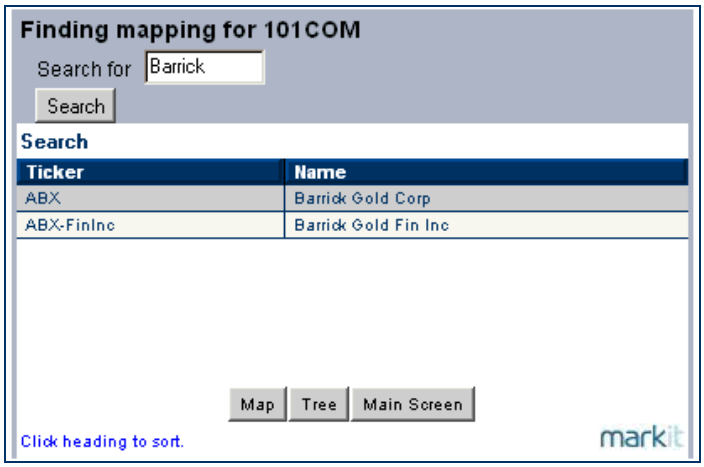
The Status column can be filtered by choosing a status code from the Status drop-down list. In this example, the Status column is not filtered; instead, it is set to All.

The **Search** button is used to search for a matching entity code in the Markit database by company long name for the selected Customer Ticker. The **Map** button is used to map selected Matched or Assumed codes (typically, status codes of Matched and Assumed are assigned the status of Matched or Assumed, respectively, when the Customer data is first imported). The **Unmap** button reverses, i.e. unmaps, a code that was previously mapped. The **Ignore** button flags the selected entity code to be ignored during the upload process and removes it from any error reporting. The **Delete** button removes the selected row from the database. The **Tree** button displays the company hierarchy for the selected Markit entity code.

Returning to our example, we want to map the 'BARIC' customer code to its equivalent in the Markit database. First, select the table row for BARIC and then click **Search**. The following screen displays.



This screen allows us to search the Markit database by long name. Enter 'barrick' (you can use all upper or all lower or mixed case letters, i.e. the search is case insensitive) into the Search for field and press Enter or click the **Search** button. (If you know it, you can enter a Markit Entity code and the system will search for an exact match, or you can enter any criteria, and the system will search for Markit long names that contain the search criteria you entered.) The screen displays a list of the tickers that match your search as shown in the example below. Select the ticker you want and click the **Map** button to map the customer code to the Markit code.



Automating XML File Transmission to Markit

After you have successfully manually transmitted data to the Markit database, the final step is to automate the process. To automate file transmission, it must be possible to perform the transmission from the command line without the need to access the web front end.

The following are the latest times for when data should be submitted:

Default Swap and Recovery Rate data	1:00AM GMT time
Loan Default Swap and Recovery Rate data	1:00AM GMT time
Bond data	2:30AM GMT time
Convertible Bond data	1:00AM GMT time

Note these times are adjusted during daylight savings time.

Friday night's data should be submitted on Friday evening or Saturday morning since the weekend can be used for upgrades to the website. Our processing of this data completes at approximately 7:00AM UK time the following morning.

Once you are automatically transmitting data to Markit.com, the status report at the end of the upload can be directed to a file and emailed internally to the relevant group of people managing the data.

FTP (File Transfer Protocol) is a common solution for automating file transmission. However, programming languages that are capable of communicating over the secure https port – for example: Visual Basic, Java, and Perl, and a package called curl – are preferable to FTP for the following reasons:

- A computer program communicating over https uses the authentication and authorization rules that are built into the web front end.
- A computer program communicating over https can take advantage of SSL encryption.
- The web page can respond immediately to a computer program and send back an error report, which is not possible with FTP.
- There is no obvious standard secure FTP package that works across operating systems.
- FTP is not secure since the login, password, and data are transmitted in clear text across the network. The FTP server is also vulnerable to hacking.

Curl Example

Let's look at the curl package first. Curl is available as source code or as a binary from the following websites:

http://curl.sourceforge.net/download.html

http://curl.haxx.se/download.html

The first step is to download the curl program and the OpenSSL packages (that enable curl to transmit information securely with SSL) appropriate for your computer from one of these websites.

Once the curl program has been successfully installed along with the SSL library, you can automate the process of transmitting the XML file to Markit as follows:

```
curl -F user=xmlupload -F password=98q34tsi –F asynch=Y -F "theFile=@prices.xml"
https://www.Markit.com/upload.jsp > key.txt
```

This command sets the username and password so that curl can log into the website, and then it sets the 'File to upload' input box on the upload screen to the name of the XML file. So continuing with our example, we are logging in as xmlupload, and transmitting a file called prices.xml.

The curl program prints output to a computer screen that would otherwise go to the web page. In the example above, the output is captured to a text file named markit.log.

Some customers experience problems when transmitting large XML files because the connection over the Internet times out before the status report is returned. It is possible to send the XML upload files in asynchronous mode. This means that you do not have to wait for the status report; instead a unique ID is immediately returned and this ID is then used by the customer to poll Markit's website for the status file.

As soon as the validation passes (around 10 seconds after the file has uploaded), a unique key is returned across the Internet (e.g. 3837). The number is the upload report key. The upload then proceeds normally. After it is finished, the normal report appears on the screen. Once you have this number – the upload report key – you are free to stop the connection and come back for the upload report.

You can get the upload report in two ways:

- Go to www.Markit.com, log in as an upload user, click on the "Report" tab, enter the upload report key into the input box, and click submit. The upload report displays in the web browser.
- Use the automatic report download mechanism, e.g.:

```
curl –F uploadReportKey=3837 –F user=xmlupload –F password=98q34tsi
https://www.Markit.com/upload.jsp
```

Either of these methods returns the report as pure text. Again, any language can be used - just HTTP POST the uploadReportKey to https://www.Markit.com/export.jsp with your username and password.

Please note that it is important that the URL argument begins with https:// (not http://).

Now, let's take a look at how we can automate the process with various programming languages.

Java Example

Here is example Java code that connects to the Markit website and transmits your XML file.

Java

```
import java.net.*;
import java.io.*;
import HTTPClient.*; // from http://www.innovation.ch/java/HTTPClient/
// Provides HTTPClient.NVPair and HTTPClient.Codecs
public class MarkitUploader {
public MarkitUploader(String host,
String user,
String password,
String theFile) throws Exception
{
String thisLine;
URL url = new URL(host);
java.net.HttpURLConnection con = (java.net.HttpURLConnection)url.openConnection();
con.setRequestMethod("POST");
con.setDoOutput(true);
NVPair formArgs[] = {new NVPair("user", user),
new NVPair("password", password),
new NVPair("asynch", "Y") };
NVPair fileArgs[] = {new NVPair("theFile", theFile) };
// will be replaced with Content-Type: etc...
NVPair httpHeaders[] = {new NVPair("dummy", "dummy")};
byte data[] = Codecs.mpFormDataEncode(formArgs, fileArgs, httpHeaders);
// If you need to specify a proxy, set the system property as follows:
// System.setProperty("https.proxyHost", "proxy.bank.com");
// System.setProperty("https.proxyPort", "8080");
// If your proxy requires (Basic) authorization, uncomment the following:
// String ppassword = "username:password"; // change these
// String encodedPassword = Codecs.base64Encode( ppassword );
// con.setRequestProperty( "Proxy-Authorization", "Basic "+encodedPassword );
// Take this line out on some networks with a proxy
con.setRequestProperty("Content-Length", ""+data.length);
con.setRequestProperty(httpHeaders[0].getName(), httpHeaders[0].getValue());
OutputStream output = con.getOutputStream();
```

```
output.write(data);
output.close();
// SSLException thrown here if server certificate is invalid
BufferedReader input = new BufferedReader(
new InputStreamReader(con.getInputStream())
);
while ((thisLine = input.readLine()) != null)
{
System.out.println(thisLine);
}
}
public static void main(String argv[]) throws Exception
{
MarkitUploader me = new MarkitUploader(argv[0], argv[1], argv[2], argv[3]);
}
}
```

The requirements for the example Java class are JDK/JRE 1.4 (which includes https support) or a previous JDK/JRE with JSSE (Java Secure Sockets Extension). Once installed, there is a potential issue concerning the relatively old Root Certificates included with the JDKs.

The Verisign Class 4 Public Primary CA certificates are located on your machine in the java.home/lib/security/cacerts file. You need to keep this file up to date with CAs you are willing to trust, i.e. this directory must include the root CA and intermediate CA (both Verisign) which signed www.Markit.com's server certificate. These certificates can be downloaded from Verisign using the following commands:

```
/usr/java/j2sdk1.4.0_01/bin/keytool -import -alias vc3ppca -keystore
/usr/java/j2sdk1.4.0_01/jre/lib/security/cacerts -storetype jks -storepass XXXXXX -file
vc3ppca.cer

/usr/java/j2sdk1.4.0_01/bin/keytool -import -alias vc3ppca2 -keystore
/usr/java/j2sdk1.4.0_01/jre/lib/security/cacerts -storetype jks -storepass XXXXXX -file
vc3ppca2.cer
```

You will need to modify the paths and provide the correct -storepass password for your CAcerts file.

Retrieving Data from the Markit Website

This section describes how to automatically retrieve data from the website.

➲ **NOTE**: Most of these reports are also available online in the Download tab of Markit.com.

Composite and cleaned contributor data is available for automated download from the Markit website once all contributor data has been submitted and all the Markit processes have completed. This should be at around 7:00AM UK time (2:00AM EST) for the previous day's closing data.

Reports can be automatically retrieved from the website in one of three formats:

- CSV – This format is useful for loading directly into Excel, but the files are difficult to parse with a computer program because the data may contain commas and double quotes, which are used as separators.
- Tab Delimited – This file format is very easy to parse by a program because none of the data contains tabs, so this character can be safely used as a delimiter.
- XML – This format is designed for reading with a SAX parser, which makes the handling of data very straightforward. A SAX parser is also able to cope with structural changes to the XML very easily.

The tables below describe which parameters you must pass in from your program to retrieve a particular report.

All Reports

The following parameters are required for all reports.

Parameter (case sensitive)	Values	Description
user	-	The user's username.
password	-	The user's password.
date	In yyyymmdd format	The date of the download file to be retrieved. This is not required for the RED and Mapping reports.
format	csv	Download in csv format.
	tab	Download in Tab delimited format.
	xml	Download in XML format.

Daily Contributor and Composite Reports

The following parameters are for retrieving the daily contributor and composite reports that correspond to the reports on the Download screen on the website. Note that as of February 23rd 2009, Markit is publishing a new fixed coupon composite report in addition to the existing daily par spreads reports. The report provides upfronts and quoted spreads on curves trading with the fixed coupon trading conventions as introduced throughout 2009.

Please see the bond and CDS column definitions for a detailed description of each report. The report columns are also described online on the Download screen.

Also note that all Current Reports are delivered in zip format.

Parameter (case sensitive)	Values	Description
report	FIXED_COUPON	A report displaying upfront and quotes spread composites. For “type” below, only use CDS
	IG_CONTRIBUTIONS	A report of investment grade contributions.
	SUB_IG_CONTRIBUTIONS	A report of sub-investment grade contributions. This is separated from the Investment Grade report since this data is subject to visibility rules.
	COMPOSITES_BY_CONVENTION	CDS-only report of composite spreads for the currency and documentation clause that is conventionally traded based on the entity’s region.
	COMPOSITES	A report of composites.
	THEORETICALS	A report of the price obtained from the theoretical credit curve. For the bond, the price is backed out from the credit curve at the bond's maturity using a probability of default model and for a default swap the price comes directly from the curve.
	DATAQUALITY	A report of the comparison of a contributor’s data against the Markit XComposite as well as measurements of the quality of the data in the form of an outlier t-test, a stale data test and a flat curve test.
	LIQUIDITY_METRICS	A report returning liquidity information (bid/ask spreads, market depth, liquidity scores) for CDS. For “typ”e below, only use CDS
type	CDS	A download containing default swap data.
	LCDS	A download containing loan default swap data.
	Bond	A download containing bond data.
version	5, for all reports other than: CDS - DATAQUALITY - 6 CDS - IG_CONTRIBUTIONS - 6 CDS - SUB_IG_CONTRIBUTIONS - 6	The latest version for these reports.

Industry Sector and Rating Reports

The following parameters are for retrieving the **industry sector and rating** levels and daily changes reports in either par or zero format.

Note that as of March 2006, Markit adopted ICB categories to generate our sector curves. The Industry Classification Benchmark (ICB) is a four-tiered system used to classify entities based on their revenue sources, i.e., sectors. The four tiers are Industry, Super-Sector, Sector, and Sub-Sector. Note that, over time, Markit will use the new ICB tier names.

Parameter (case sensitive)	Values	Description
report	sector	The aggregated risk reports.
sectortype	levels	An industry sector curves download of curve levels.
parzero	Par	The industry sector curve levels are expressed in par format.
	Zero	The industry sector curve levels are expressed in zero coupon format.
version	7	The latest version of this report.

CDS Sameday Reports

The following parameters are for retrieving the daily **CDS Sameday** composite reports. These reports are intra-day updates for the CDS market made available at the close of each primary time zone. As with the daily composite reports, the data is delivered in Zip format.

Parameter (case sensitive)	Values (case insensitive)	Description
report	Same_Day_CDS_New_York	A report of CDS updated for and available at NY market close.
	Same_Day_CDS_London	A report of CDS updated for and available at London market close.
	Same_Day_CDS_Europe	A report of CDS updated for and available at Europe market close.
	Same_Day_CDS_London_Midday	A report of CDS updated for and available at London Midday.
	Same_Day_CDS_Asia	A report of CDS updated for and available at Asia market close
	Same_Day_CDS_Japan	A report of CDS updated for and available at Japan market close
type	CDS	A download containing default swap data
version	6	The latest available version of these reports.

LCDS Sameday Reports

The following parameters are for retrieving the daily **LCDS Sameday** composite reports. These reports are intra-day updates for the CDS market made available at the close of each primary time zone, i.e. Asia, London, and New York. As with the daily composite reports, the data is delivered in Zip format.

Parameter (case sensitive)	Values (case insensitive)	Description
	Same_Day_LCDS_New_York	A report of LCDS updated for and available at NY market close.

Parameter (case sensitive)	Values (case insensitive)	Description
	Same_Day_LCDS_London	A report of LCDS updated for and available at London market close.
	Same_Day_LCDS_Europe	A report of LCDS updated for and available at Europe market close.
	Same_Day_LCDS_Tokyo	A report of LCDS updated for and available at Tokyo market close.
Type	LCDS	A download containing loan default swap data.
version	5	The latest available version of these reports.

Note: Same Day LCDS Composite reports for all regions to include Legacy and Bullet Composites

RED Entity and Obligation, Credit Index Annex, and Corporate Actions Reports

The following parameters are for retrieving the **RED Entity and Obligation** (full or unmapped and mapped versions) reports, as well as, the **Credit Index Annex** and **Corporate Actions** reports. Note that this report does not take a date parameter.

Parameter (case sensitive)	Values	Description
report	REDEntity	A download of the entities in the RED database.
	REDObligation	A download of the obligations in the RED database.
	REDEntityMapped	A download for RED By-mappings customers of the entities in the RED database for which the client has mappings.
	REDObligationMapped	A download for RED By-mappings customers of the obligations in the RED database for which the client has mappings.
	REDEntityDelta	A download of the changes in the RED Entities XML for the past five business days.
	REDObligationDelta	A download of the changes in the RED Obligations XML for the past five business days.
	CredIndexAnnex	A download of the CDX and iTraxx credit index annexes.
	REDCorporateAction	A download of all the corporate actions implemented.

Parameter (case sensitive)	Values	Description
family (for CredIndexAnnex report value only)	ITRAXX-ASIA ITRAXX-EUROPE CDX MCDXNA “IBOXX Index“ “TRAC-X Asia Index“ “TRAC-X Europe Index“ “TRAC-X NA Index“ ITRAXX-SDI ITRAXX-SOVX	The index family to retrieve for the CredIndexAnnex report (used for the CredIndexAnnex value only).
version	7 for REDCorporateAction 8 for REDEntity & REDEntityDelta 6 for all other reports	The latest available version of these reports.

Credit Index Composites/Theoretical and Sameday Reports

The following parameters are for retrieving a download of **credit index** composite and theoretical spread and prices, and Sameday Index reports.

Parameter (case sensitive)	Values	Description
report	COMPOSITES	A download of the credit index composite and theoretical price and spreads. Theoretical prices and spreads are dependant on permissioning.
	CONSTITUENTS_COMPOSITES	A download of composite prices for the single name default swaps underlying the index.
	TRANCHE_COMPOSITES	A download of the composite spreads and upfronts on credit index tranches.
	SAME_DAY_INDICES	A download of the same day credit index composite and theoretical price and spreads. Theoretical prices and spreads are dependant on permissioning.
family	ITRAXX-ASIA ITRAXX-EUROPE CDX MCDXNA “IBOXX Index“ “TRAC-X Asia Index“ “TRAC-X Europe Index“ “TRAC-X NA Index“ ITRAXX-SDI ITRAXX-SOVX	The index family to retrieve. If you are permissioned for an index family, the download will include theoretical values. Otherwise, an unpermissioned version of this report is provided without theoretical values.
type	Credindex	Specifies that this is a credit index report.

Parameter (case sensitive)	Values	Description
version	4 5 for CONSTITUENTS_COMPOSITES	The latest available version of these reports.

Loan Credit Index Composites Reports

The following parameters are for retrieving a download of **loan credit index** composite prices.

Parameter (case sensitive)	Values (case insensitive)	Description
report	COMPOSITES	A download of the credit index composite and theoretical price and spreads. Theoretical prices and spreads are dependant on permissioning.
family	LEVX LCDX	The index family to retrieve.
type	Loancredindex	Specifies that this is a loan credit index report.
version	4	The latest available version of these reports.

RED Loans Reports

The following parameters are for retrieving a download of **RED Loans** report, as well as the **Loan Credit Indices Annexes** reports. Note that these reports do not require a data parameter.

Parameter (case sensitive)	Values	Description
report	redloanssl	A download of the of the entities and related obligations in the RED LCDS North American Service
	redloanseurope	A download of the of the entities and related obligations in the RED LCDS European Service
	LOANSINDEXANNEX	A download of the LCDX and LevX Loan Credit Index Annexes
family (for LOANSINDEXANNEX report values only)	LCDX “iTraxx LevX”	The index family to retrieve.
version	6	The latest available version of these reports.

Entity Mapping Report

The following parameters are for retrieving a download of the **entity mappings** screen. Note this report does not take a date parameter.

Parameter (case sensitive)	Values (case insensitive)	Description
report	mappings	A download of the entity mappings screen.
version	2	The latest available version of this report.

Structured Finance Index Components Reports

Parameter (case sensitive)	Values (case insensitive)	Description
report	sfxcomponents	This report displays the components (or constituent names) of structured finance indices.
family	ABX.HE CMBX	Home equity and commercial mortgage-backed families are available.
type	abscredidx	Structured finance index.
version	5	The latest available version of this report.

For sample code to retrieve this report, see the Curl section.

Structured Finance Index Composites Reports

Parameter (case sensitive)	Values (case insensitive)	Description
report	xcomposites	This report displays composite pricing for structured finance indices.
family	ABX.HE CMBX	Home equity and commercial mortgage-backed families are available.
type	abscredidx	Structured finance index.
version	5	Latest available version of this report.

For sample code to retrieve this report, see the Curl section.

Convertibles Composites Reports

The Convertible API includes the following fields: ISIN, Bid, Offer, Equity Price, Delta, Depth, Tier, Rating, Currency, Region, and Sector. This information is also found on the download tab on the website and is available in csv, xml, and tab.

Parameter (case sensitive)	Values (case insensitive)	Description
report	composites	This report displays composite pricing for structured finance indices.
type	convert	Structured finance index.
version	4	Latest available version of this report.

For sample code to retrieve this report, see the Curl section.

Automated Retrieval Samples

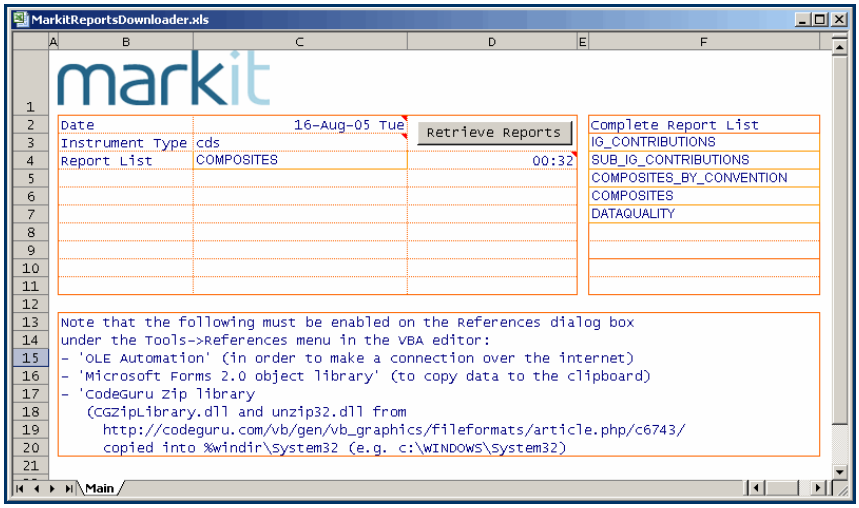
Excel Visual Basic

Here is an example Visual Basic program for automating the download, which was developed and tested in MS-Excel 2000.

Please note that the following must be selected on the References dialog box under the Tools > References menu in the VBA editor:

- OLE Automation (to connect to the website over the Internet)
- Microsoft Forms 2.0 object library (to hold data in the clipboard)
- Microsoft XML, version… >= 5

The example code works alongside a sheet that has the 'Date', 'Instrument_Type', 'WinZip' and 'Report_List' (C4:C9) named ranges defined, as in the following example screen.



Visual Basic

```
Option Explicit

' These are global variables used by the GetPassword() function.
' The password and username variables are global so that the values entered are
' remembered between calls to GetPassword().
' The globalCancel variable is global so that it can flag to GetPassword() when
' Escape or the Cancel button was pressed.
Public globalCancel As Boolean, password$, username$


' Note: Enable the following in the References dialog box under the Tools > References menu
' in the VBA editor:
' - 'OLE Automation' (to make a connection over the Internet)
' - 'Microsoft Forms 2.0 object library' (to copy data to the clipboard)
Sub GetMarkitData()
    Dim doc 'As MSXML2.ServerXMLHTTP   ' For storing the downloaded report
    Dim clipboard As DataObject  ' For storing the report in the clipboard to paste into the
spreadsheet
    Dim s$                       ' The string sent to the server for retrieving the report
    Dim w As Worksheet           ' Used when searching for a sheet with a name that corresponds
with a report
    Dim i%                       ' For looping through each report listed on the 'Main' sheet
    Dim timetaken#               ' Record how long the download took to complete
    Dim direc$                   ' Where to store the downloaded ZIP file (for version=5)
    Dim filenumber#              ' For writing to the ZIP file
```

```
    Dim myfile                  ' For iterating through the ZIP file

    ' Pop up a password input box
    If Not GetPassword(username, password) Then
        Exit Sub
    End If

    ' Set the 'Data -> Text to Columns' menu setting to CSV (comma delimited) format which
    ' corresponds with the csv setting when downloading from the Markit website. Excel will
    ' then automatically put the download into separate cells.
    ActiveSheet.Range("Report_List").TextToColumns Destination:=ActiveSheet.Range("Report_List"),
_
        DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote, _
        ConsecutiveDelimiter:=False, _
        Tab:=False, _
        Semicolon:=False, _
        Comma:=True, _
        Space:=False, _
        Other:=False

    ' Loop through the 'Report List' range on the Main sheet.
    For i = 0 To 100

        ' Open a connection to the Internet.
        ' Note that 'OLE Automation' on in the References dialog box (under the Tools->References
        ' menu in the VBA editor) must be enabled for the following to work
        Set doc = CreateObject("msxml2.ServerXMLHTTP")
        doc.Open "POST", "https://www.Markit.com/export.jsp", False
        doc.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"

        ' If you need to tell VB about your proxy server, you should change ServerXMLHTTP
        ' above to ServerXMLHTTP.4.0 or ….5.0 (depending on your version of MSXML) and
        ' add:
        ' doc.setProxy 2, "proxy.bank.com:8080" ' change me
        ' doc.setProxyCredentials "username", "password" ' change me

        ' Turn off SSL cert checking with this:
        'doc.setOption 2, doc.getOption(2) - SXH_SERVER_CERT_IGNORE_ALL_SERVER_ERRORS

        ' Have we reached the end of the Report List on the Main sheet
        If Range("Report_List").Offset(i) = "" Then Exit For

        ' Start timing
        timetaken = Now()

        ' The type parameter specifies either 'bond' or 'cds' instrument type
        s = "user=" & username _
          & "&password=" & password _
          & "&version=" & "5" _
          & "&date=" & Format(Range("Date"), "yyyymmdd") _
          & "&format=" & "csv" _
          & "&report=" & Range("Report_List").Offset(i) _
          & "&type=" & Range("Instrument_Type")
```

```
        ' Send the request to the Markit website
        doc.send (s)

        ' Create or find a sheet for storing the downloaded report with the same name
        ' as that report.
        For Each w In Worksheets
            If w.Name = Range("Report_List").Offset(i) Then Exit For
        Next
        If w Is Nothing Then
            Set w = Sheets.Add(after:=Sheets(Sheets.Count))
            w.Name = Range("Report_List").Offset(i)
        Else
            w.Activate
        End If

        ' Handle the response
        If Left(doc.getResponseHeader("Content-Type"), 17) = "application/x-zip" Then ' A ZIP file

            ' Save version=5 zip file to a directory
            ' requires WinZip or similar utility to be installed

            On Error Resume Next
            direc = "c:\temp\Markit"
            MkDir direc
            On Error GoTo 0
            direc = direc & "\d" & (Year(Now) * 10000000000# + _
                                    Month(Now) * 100000000 + _
                                    Day(Now) * 1000000 + _
                                    Hour(Now) * 10000 + Minute(Now) * 100 + Second(Now))
            MkDir direc
            Dim inputFile() As Byte
            inputFile = doc.responseBody

            filenumber = FreeFile
            m body1() As Byte
            Dim body2() As Byte

            body1 = doc.responseBody
            ReDim body2(UBound(body1))
            body2 = body1

            Open direc & "\file.zip" For Binary Access Write As #filenumber
            Put #filenumber, 1, inputFile
            Close #filenumber
            Set doc = Nothing

            ' Use winzip to unzip it
            Shell ("""c:\program files\winzip\winzip32.exe"" -e -o """ & direc & "\file.zip"" """ & direc & """")


Shell """" & Range("WinZIP") & """ -e -o """ & direc & "\file.zip"" """ & direc & """", vbHide = True
            ' -e extract
            ' -o overwrite
```

```
            ' Give winzip time to do its stuff
            Application.Wait (Now + TimeValue("0:00:5"))
            ' Open unzipped file in a new workbook
            myfile = Dir(direc & "\*")
            Do While myfile <> ""
                If myfile <> "." And myfile <> ".." And Right$(myfile, 4) <> ".zip" And myfile <>
"Disclaimer.txt" Then
                    'This must be the .csv file contained in the ZIP file

                    w.AutoFilterMode = False
                    w.Cells.Clear
                    With w.QueryTables.Add(Connection:="URL;" & "file:" & direc & "\" & myfile,
Destination:=w.Cells(1, 1))
                        .BackgroundQuery = False
                        .Refresh
                        .Delete
                    End With

                    w.Select
                    Columns(1).Select
                    Selection.TextToColumns DataType:=xlDelimited, ConsecutiveDelimiter:=True,
Comma:=True, Space:=False
                End If
                myfile = Dir
            Loop
    Else
        ' Now paste the report into the sheet.
        ActiveSheet.Cells.Clear
        ActiveSheet.Range("A1").Select
        Set clipboard = New DataObject
        clipboard.SetText doc.responseText
        clipboard.PutInClipboard
        ActiveSheet.Paste
    End If

    ' Record the time taken to retrieve the report.
    Range("Report_List").Offset(i, 1) = Now() - timetaken

    Next

End Sub

' This function creates a new Workbook and DialogSheet within the Workbook on which
' the username and password are prompted.
Function GetPassword(ByRef username$, ByRef password$) As Boolean
    Dim thisW As Workbook   ' Remember the current workbook
    Dim w As Workbook       ' The new workbook for putting the password input box onto
    Dim d As DialogSheet    ' The dialogsheet for putting the password input box onto
    Dim o

    Set thisW = ActiveWorkbook
    Application.ScreenUpdating = False
    Set w = Workbooks.Add
    Set d = DialogSheets.Add
    d.DialogFrame.Characters.Text = "Markit Password Input Box"
```

```
    For Each o In d.DrawingObjects: o.Delete: Next

    ' Left, Top, Width, Height
    d.Labels.Add(80, 60, 150, 12).Characters.Text = "Username"
    With d.EditBoxes.Add(180, 60, 100, 12)
        .InputType = xlText
        .Text = username
    End With

    d.Labels.Add(80, 80, 150, 12).Characters.Text = "Password"
    With d.EditBoxes.Add(180, 80, 100, 12)
        .InputType = xlText
        .PasswordEdit = True
        .Text = password
    End With

    With d.Buttons.Add(80, 120, 50, 12)
        .Characters.Text = "OK"
        .DefaultButton = True
        .CancelButton = False
        .DismissButton = True
    End With

    With d.Buttons.Add(140, 120, 50, 12)
        .Characters.Text = "Cancel"
        .DefaultButton = False
        .CancelButton = True
        .DismissButton = False
        .OnAction = "EscapePressed"
    End With

    thisW.Activate
    Application.ScreenUpdating = True
    globalCancel = False
    d.Show

    username = d.EditBoxes(1).Text
    password = d.EditBoxes(2).Text

    w.Close savechanges:=False
    GetPassword = Not globalCancel

End Function

' The Escape key or Cancel button was pressed so record in the 'globalCancel' variable.
Sub EscapePressed()
globalCancel = True
End Sub
```

Java

Here is an example Java program for automating the download.

Java

```
import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.SocketTimeoutException;
import java.util.zip.ZipInputStream;
import org.apache.commons.httpclient.HttpException;
import org.apache.commons.httpclient.NameValuePair;
import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.methods.PostMethod;
public class ExampleDownloader {
private static HttpClient client = new HttpClient();
public enum reportResponse {SUCCESS, ALREADYEXISTS, DBBUSY, BUILDING, LOGINERROR, NOPERMISSION,
FAILURE, TIMEOUT};
public static void main(String[] arg) {
ExampleDownloader.downloadReport(
"lcds", // type
"composites", // report
5, // version
"xml", // format
"http://www.markit.com/export.jsp", // host
"bob", // username
"xxxx", // password
"19Jun08", // date
"Reports"); //output folder
}
/**
* Download a report from Markit
* @param report
* @param format
* @param host
* @param user
* @param password
* @param date
* @param output
* @return whether or not the download succeeded
*/
public static reportResponse downloadReport(String type, String report, int version, String
format, String host, String user, String password, String date, String output) {
//Create post method to send to server
PostMethod method = new PostMethod(host);
NameValuePair nvpuser = new NameValuePair("user", user);
NameValuePair nvppassword = new NameValuePair("password", password);
NameValuePair nvpdate = new NameValuePair("date", date); // yyyymmdd format
NameValuePair nvpformat = new NameValuePair("format", format); // xml, csv or tab
NameValuePair nvptype = new NameValuePair("type", type);
NameValuePair nvpreport = new NameValuePair("report", report);
NameValuePair nvpfamily = new NameValuePair("family", null);
NameValuePair nvpversion = new NameValuePair("version", String.valueOf(version));
NameValuePair[] nvpArray = new NameValuePair[] {nvpuser, nvppassword, nvpdate, nvpformat,
nvptype, nvpreport, nvpfamily, nvpversion};
method.setRequestBody(nvpArray);
//If there is no response from the server in this time then the report request times out
method.getParams().setSoTimeout(60 * 1000);
//Generate a unique filename based on the parameters
String fileName = type + "_" + report + "_" + version + "_" + user + "_" + date + "." +
format;
//Add a slash to the output folder if it doesn't already have one
if (!output.endsWith("/") || !output.endsWith("\\")) {
output += "/";
}
File outputFile = new File(output + fileName);
try {
int statusCode = client.executeMethod(method);
if (statusCode == 200) {
//Request succeded
try {
InputStream is = method.getResponseBodyAsStream();
//If the response is actually a zip then create a ZipInputStream to
//read the response body
String contentTypeHeader = method.getResponseHeader("Content-Type").getValue();
if (contentTypeHeader.startsWith("application/x-zip")) {
//This is a zip
//isZip = true;
ZipInputStream zis = new ZipInputStream(new BufferedInputStream(is));
//Skip the first Disclaimer.txt entry
zis.getNextEntry();
//Skip the report entry
zis.getNextEntry();
is = zis;
```

```
}
//Make sure the output folder actually exists
if (!outputFile.getParentFile().exists()) {
if (!outputFile.getParentFile().mkdirs()) {
System.err.println("Error creating output folder: " + outputFile.getParent());
}
}
//Write the response to a file
FileOutputStream fos = new FileOutputStream(outputFile);
BufferedWriter wr = new BufferedWriter(new FileWriter(outputFile));
BufferedReader rd = new BufferedReader(new InputStreamReader(is));
//To store the response as a string so we can check for error messages
String responseText = ""; //new String();
int totalBytesRead = 0;
String line;
while ((line = rd.readLine()) != null) {
totalBytesRead += line.length();
//Output the first chunk of the response to a string
if (responseText.length() <= 350) {
responseText += line + "\r\n";
}
wr.write(line);
//This will probably not be the new line char used by the original report
//but this shouldn't matter for comparisons with a diff tool
wr.newLine();
}
wr.flush();
fos.close();
is.close();

//Output the response text
if (responseText.indexOf("Database server is busy") >= 0) {
return reportResponse.DBBUSY;
} else if (responseText.indexOf("Error logging in") >= 0) {
return reportResponse.LOGINERROR;
} else if(responseText.indexOf("This report is currently building") >= 0) {
return reportResponse.BUILDING;
} else if (responseText.indexOf("You are not permissioned") >= 0) {
return reportResponse.NOPERMISSION;
} else {
System.out.println("Your report has been downloaded to " + fileName);
return reportResponse.SUCCESS;
}
} catch (SocketTimeoutException e) {
return reportResponse.TIMEOUT;
}
}
} catch (HttpException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
} finally {
method.releaseConnection();
}
return reportResponse.FAILURE;
}

}
```

Curl

Here is an example Curl program for automating the download of bond composites:

Curl

```
curl -F user=$user -F password=$password -F date=$yyyymmdd -F format=tab -F report=Composites -F
version=5 -F type=bond https://www.Markit.com/export.jsp > Composites.zip
```

The curl sample below may be used to retrieve Structured Finance Index Components:

Curl

```
curl -k -F username=$user -F password=$password -F date=$yyyymmdd -F format=xml -F
report=sfxcomponents -F family=ABX.HE -F type=abscrediдx -F version=5
https://www.markit.com/export.jsp > abx_components.zip
```

The curl sample below may be used to retrieve Structured Finance Index Composites:

Curl

```
curl -k -F username=$user -F password=$password -F date=$yyyymmdd -F format=xml -F
report=xcomposites -F family=ABX.HE -F type=abscredidx -F version=5
https://www.markit.com/export.jsp > abx_composites.zip
```

The curl sample below may be used to retrieve Convertibles Composites:

Curl

```
curl -F username=Username -F password=Password  -F date=20060919  -F format=csv -F
report=COMPOSITES -F type=convert -F version=5 https://www.markit.com/export.jsp >
ConvertibleMarket.zip
```

The curl sample below may be used to retrieve Loans Default Swap data:

Curl(Version 5)

```
curl  -F username=******  -F password=******   -F format=csv -F type=lcds  -F report=COMPOSITES  -
F version=5 -F date=20070109 -FdoNotGzip=true https://www.markit.com/export.jsp  > lcds.zip
```

Curl(Version 6)

```
curl  -F username=******  -F password=******   -F format=csv -F type=lcds  -F report=
BULLET_COMPOSITES  -F version= 6 -F date=20070109 –FdoNotGzip = true
https://www.markit.com/export.jsp  > lcds.zip
```

To download the LCDS reports through the download api, use exactly the same parameters as the CDS composites report, except change the type to "LCDS" from "CDS".

Additional Resources

This section provides information about how to contact Markit for more information or support and how to obtain online help.

➲ **NOTE**: Access the companion guide to this guide, the Markit.com User Guide, on the Markit.com website under **Help** or contact your sales representative to request copies. See the following two sections, Contact Us and Online Help, for details.

Contact Us

For questions not answered in this guide, please contact the Markit Client Relations:

- support@markit.com
- Phone

Region	Telephone
Europe	+800 6275 4800
America	+1 877 762 7548
Asia	+65 6499 0079
Japan	+81 3 6402 0127

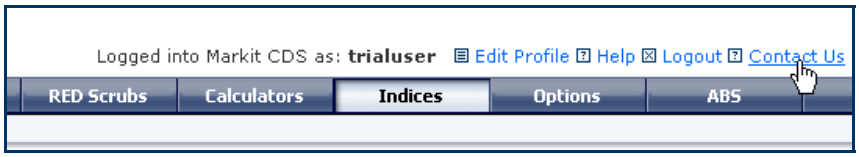
For information about Markit's other services, please contact a Markit sales representative by:

- email at sales@markit.com, or
- For full contact information click here

Online Help

Help is available online when you click the **Help** link in the Markit.com header, as shown below. On the Help page, you can access the companion Markit.com User Guide, a list of Frequently Asked Questions (FAQ), Release Notes (which describe the latest Markit.com enhancements), and other useful documents.

As shown in the following graphic, click **Contact Us** to send an email message directly to the Markit Client Relations Team.



Appendix A – Error Codes during XML Upload

Errors that begin with 'CRITICAL' halt the upload process immediately. Other errors apply to the line in the XML file referred to.

Errors that begin with 'curl:' are due to an incorrect parameter passed to curl. For example, 'curl: (26) failed creating formpost data' is caused when the value assigned to 'theFile' is a file that does not exist.

A 'Login Failed' error when trying to transmit data with curl may result from an URL argument that does not begin with 'https://'.

Error Message	Description
`The derivatives database is unavailable at present. Please try again later.`	The database is off-line.
`Error in upload file, line 84, char 20 Message: Attribute 'type' does not match its defined enumeration or notation list`	An attribute to one of the tags has been set to an incorrect value on the line specified in the error message. This error may be for the <bondid>, <price> or <spread> tags, which each take a 'type' attribute.
`Fatal Error in upload file, line 7, char 3 Message: Expected an element name`	This error occurs when an XML tag has a space between the opening or closing bracket and the tag name at the line specified in the error message.
`Error in upload file, line 9, char 153 Message: Unknown element 'position'`	An unknown tag was found within the XML file at the line specified in the error message. In this case the <position> tag was spelled incorrectly.
`Fatal Error in upload file, line 198, char 1 Message: The input ended before all started tags were ended. Last tag started was 'import'`	A closing tag is missing at the line specified in the error message. In this case the final </import> tag that appears at the end of the XML file is missing.
`Fatal Error in upload file, line 2, char 62 Message: An exception occurred! Type:NetAccessorException, Message:The host/address '(0)' could not be resolved. scanFirst() failed`	The URL for Markit was not correctly specified in the DOCTYPE tag at the top of the XML file. It is important that this line reads: <!DOCTYPE import SYSTEM "http://www.mark-it.com/import.dtd">
`Fatal Error in upload file, line 10, char 31 Message: Expected end of tag 'entity'`	A tag was opened, but a corresponding closing tag was not found, in this case the closing entity tag </entity> was not found on line 10 of the XML file
`Unknown bond, description 'LAND BRANDENBURG 04/24/2009 5.25' and isin 'DE0001076388'`	A customer has transmitted a price to Markit for a bond that was not found in the Markit database. This error message is improved with a bond description if the <bondname> XML tag is supplied with a bond name.
`Entity BNROMA is within a hierarchy. The curve will not be inserted until the mapping is confirmed.`	The customer entity code was found in the Markit database; however, it must be mapped on the Markit Entity screen before it is uploaded into the Markit database.
`Entity CRDSUI has a match in data element 255. The curve will not be inserted until the mapping is confirmed.`	The customer entity code was found in the Markit database; however, it must be mapped on the Markit Entity screen before it is uploaded into the Markit database.

Error Message	Description
Entity GGX flagged as de-merged. Ignoring data element 19.	This customer entity code is mapped to a Markit code that has been de-merged, and Markit have updated their database to reflect this. This entity will be flagged as 'De-Merged' in the Status column of the Markit Entity screen, and must be re-mapped.
Entity RKCC flagged as bankrupt. Ignoring data element 2.	This customer entity code is mapped to a Markit code that has been flagged as bankrupt in the Markit database.
Unable to find match for entity RNBRK in data element 4.	The customer entity code does not match a code within the Markit database. This code will be flagged as 'Not Found' in the Status column of the Markit Entity screen, and must be mapped to the correct entity within the Markit database before default swap data for this reference entity feeds in.
ERROR: BondMtm(): Invalid bond price low=-.029 high=-.029 for Instrument "ESKOM 0 31Dec32", ISIN=DS8834723423, Cusip=TT333904 for book ASWP	An attempt was made to feed an invalid bond price into the Markit database. Bond prices may not be zero or negative.
ERROR: EK SNRFOR 30Y Default. ORA-02290: check constraint (MARKIT.MTMNOZEROS) violated	An attempt was made to feed a default swap spread of zero into the Markit database. Default swap spreads may not be zero or negative.
Duplicate curve provided for 'PGNIG SNRFOR'. Curve will not be inserted for data element 194.	The XML file contains two <credcurve> elements that refer to the same curve. The first one that was encountered will have been entered into the Markit database. The second one will not be entered into the database.
Must provide at least one valid delta in creddelta at data element 1	A <creddelta> tag does not contain any delta tags, i.e. it is empty.
Must provide at least one valid spread in a credcurve element at data element 123.	An attempt was made to feed a <credswap> XML tag that contained no spread information. At least one of the spread tags must be supplied.
System 'Global' is not recognised. Please contact Markit to agree system names	The <system> tag within the XML file refers to a system that has not been agreed with Markit. Please contact Markit so that the details of this system can be entered into the Markit database.
System 'Global' does not have a fileid of 2. Please contact Markit to agree a new fileid.	The <fileid> tag within the XML file refers to a fileid that has not been agreed with Markit. Please contact Markit so that this fileid can be entered into the Markit database.
AX - Date '26-Jun-2002' provided in header record is not today's date '27-JUN-2002'. Contact Markit if you wish to upload backdated data.	Data will only be inserted automatically through the web for today's date. The data will be stored in a holding area, and can be entered into the database by Markit; however, this would involve re-running the data quality, credit curve, sector curve and theoretical calculations.

Error Message	Description
`Cannot perform upload at present. It is already running elsewhere.`	An upload to Markit was started while another upload was already running for that system and fileid. Wait until the other upload has completed.
`Fatal Error in upload file, line 2, char 18 Message: Expected system or public id. scanFirst() failed`	It is important that SYSTEM appears in upper case in the DTD line: <!DOCTYPE import SYSTEM "http://www.mark-it.com/import.dtd">
`Fatal Error in upload file, line 2, char 62 Message: An exception occurred! Type:NetAccessorException, Message:Could not open file: {0}. scanFirst() failed.`	The filename for the DTD was not correctly specified in the DOCTYPE tag at the top of the XML file. It is important that this line reads: <!DOCTYPE import SYSTEM "http://www.mark-it.com/import.dtd">
`Time '18:30' is not a valid time. Must be in format HH:MM TZN in data element 1.`	The time must be in 24 hour format with a time zone string.

Appendix B – Error Messages during Download

The following table lists the error messages that can be returned when experiencing issues with CURL-based downloads.

Error Message	Description
“A report name must be specified for vX downloads”	There is no report name being specified in the CURL script.
“Error logging in”	The username/password combination being used in the CURL script is incorrect.
“Report is not available for this date”	There is no report available for the date being specified
“Could not validate date”	The date format being used is incorrect
“Unknown report arguments”	The report arguments being specified are either in an incorrect format or do not exist.
“Error generating version X report”	The version being specified does not exist for the report being requested
"Unable to perform download. Batch is currently running."	The overnight batch is currently running and therefore the report is currently not available.
"Database server is busy. Please try again later."	The server is currently too busy to process the request. Wait a few minutes and try again.
"This report is currently building, please try again in a few minutes"	The report has been requested for a second time before the first request has been completed.
“You are not permissioned to see CDS. Please contact Markit”	There is no report name being specified in the CURL script.
“You are not permissioned to see composites. Please contact Markit”	The username/password combination being used in the CURL script is incorrect.
“You are not permissioned to see XXX Sameday data. If you would like to be permissioned for Markit Sameday data, please contact info@markit.com”	You are not permissioned to see the Sameday data being requested.
“XXX data has not yet been released; please contact Markit”	The Sameday data for the region being specified has not been released.

Appendix C – Unzipping a Download File in Visual Basic

Visual Basic

```
' Save version=5 zip file to a directory
' requires WinZip or similar utility to be installed

On Error Resume Next
direc = "c:\temp\Markit"
MkDir direc
On Error GoTo 0
direc = direc & "\d" & (Year(Now) * 10000000000# + Month(Now) * 100000000 + Day(Now) * 1000000 + Hour(Now) * 10000 + Minute(Now) * 100 + Second(Now))
MkDir direc

filenumber = FreeFile
Open direc & "\file.zip" For Binary Access Write As #filenumber
Dim arr1() as Byte
arr1 = xmlDoc.responseBody
Put #iFileNum, 1, arr1
Close #filenumber

Set xmlDoc = Nothing

' Use winzip to unzip it
ID = Shell("""c:\program files\winzip\winzip32.exe"" -e -o ""c:\temp\file.zip"" """ & direc & """")
' Give winzip time to do its stuff
Application.Wait (Now + TimeValue("0:00:5"))

' Open unzipped file in a new workbook
myfile = Dir(direc & "\*")
Do While myfile <> ""
    If myfile <> "." And myfile <> ".." And Right$(myfile, 4) <> ".zip" Then
        Workbooks.Open Filename:=direc & "\" & myfile, ReadOnly:=True
    End If
    myfile = Dir
Loop
```

Appendix D – Obtaining the Time Zone

In UNIX, it is possible to construct the <time> tag in the correct format with the time zone with the date command using this format:

```
date '+%H:%M %Z'
```

In Windows, with Visual Basic, use the following example code to provide the time zone, which builds the <time> tag in the correct format:

```
Private Type SYSTEMTIME
   wYear As Integer
   wMonth As Integer
   wDayOfWeek As Integer
   wDay As Integer
   wHour As Integer
   wMinute As Integer
   wSecond As Integer
   wMilliseconds As Integer
End Type

Private Type TIME_ZONE_INFORMATION
   Bias As Long
   StandardName(0 To 63) As Byte
   StandardDate As SYSTEMTIME
   StandardBias As Long
   DaylightName(0 To 63) As Byte
   DaylightDate As SYSTEMTIME
   DaylightBias As Long
End Type

Private Declare Function GetTimeZoneInformation _
   Lib "kernel32" (lpTimeZoneInformation As _
   TIME_ZONE_INFORMATION) As Long

Function timezone()
   Dim tz As TIME_ZONE_INFORMATION
   Select Case GetTimeZoneInformation(tz)
   Case 1
      timezone = "GMT"
   Case 2
      timezone = "BST"
   Case Else
      timezone = "UNKNOWN"
   End Select
End Function
```

Appendix E – Report Column / XML Tag Definitions

This section provides a detailed description of the column definitions for the Markit download reports.

CDS Report Columns

The following table describes the columns for the CDS reports.

Field	Description	CDS Composite	CDS Contributions	CDS Data Quality	CDS Fixed Coupon	Type
Assumed Recovery	The recovery level used to convert the Upfront value to a Conventional spread				Y	
AvRating	The average of Moody's and S&P ratings adjusted to instrument's seniority and rounded to not include '+' and '–' levels.	Y	Y	Y	Y	Char 3
Buildable	A Y/N flag to indicate whether a contributed curve is buildable			Y		
CarryFwd	Indicates the number of days a contribution has been carried forward. A price is carried forward by adjusting for the average change in all good points from one day to the next.			Y		Int
CarryFwd5y	See 'CarryFwd'		Y			Int
Ccy	The currency of the instrument.	Y	Y	Y	Y	Char 3
CleaningPriceType	The price type used for the data cleaning process		Y	Y		
CompositeDepth	The composite depth for the corresponding entity, tier, maturity, doc clasue, and currency combination			Y		
CompositeDepth5y	The number of distinct contributors at the composite fallback level.	Y	Y		Y	Int
CompositeLevel6m	The fallback level of the composite calculation.	Y			Y	Char 12
CompositeLevel1y	Can be one of CcyGrp, DocAdj, EntityTier, or Thin.	Y			Y	Char 12
CompositeLevel2y	See table of Composite Fallback Levels below	Y			Y	Char 12
CompositeLevel3y		Y			Y	Char 12
CompositeLevel4y		Y			Y	Char 12
CompositeLevel5y		Y			Y	Char 12

CompositeLevel7y		Y			Y	Char 12
CompositeLevel10y		Y			Y	Char 12
CompositeLevel15y		Y			Y	Char 12
CompositeLevel20y		Y			Y	Char 12
CompositeLevel30y		Y			Y	Char 12
CompositeCurveRating	A quantitative measure of the quality of our CDS data, averaged over the points on this curve.			Y	Y	Var char2
ContributedCoupon	The coupon used by the contributor to calculate their contributed upfront value.		Y	Y		
ContributedLevel	The level contributed for the corresponding tier, ccy, doc clause, and maturity			Y		
ContributedLevel6m	The level contributed for the 6m tenor		Y			
ContributedLevel1y	The level contributed for the 1y tenor		Y			
ContributedLevel2y	The level contributed for the 2y tenor		Y			
ContributedLevel3y	The level contributed for the 3y tenor		Y			
ContributedLevel4y	The level contributed for the 4y tenor		Y			
ContributedLevel5y	The level contributed for the 5y tenor		Y			
ContributedLevel7y	The level contributed for the 7y tenor		Y			
ContributedLevel10y	The level contributed for the 10y tenor		Y			
ContributedLevel15y	The level contributed for the 15y tenor		Y			
ContributedLevel20y	The level contributed for the 20y tenor		Y			
ContributedLevel30y	The level contributed for the 30y tenor		Y			
ContributedPriceType	The price-type contributed by the contirbutor		Y	Y		
Contributor	If the contribution is yours, this field displays ‘Yours’; otherwise, it is blank.	Y	Y		Y	Char 10
ConvSpread6m	The conventional spread level for the 6m tenor. Converted from the				Y	

	par spread.					
ConvSpread1y	The conventional spread level for the 1y tenor. Converted from the par spread.				Y	
ConvSpread2y	The conventional spread level for the 2y tenor. Converted from the par spread.				Y	
ConvSpread3y	The conventional spread level for the 3y tenor. Converted from the par spread.				Y	
ConvSpread4y	The conventional spread level for the 4y tenor. Converted from the par spread.				Y	
ConvSpread5y	The conventional spread level for the 5y tenor. Converted from the par spread.				Y	
ConvSpread7y	The conventional spread level for the 7y tenor. Converted from the par spread.				Y	
ConvSpread10y	The conventional spread level for the 10y tenor. Converted from the par spread.				Y	
ConvSpread15y	The conventional spread level for the 15y tenor. Converted from the par spread.				Y	
ConvSpread20y	The conventional spread level for the 20y tenor. Converted from the par spread.				Y	
ConvSpread30y	The conventional spread level for the 30y tenor. Converted from the par spread.				Y	
Country	The country of the issuing organization.	Y	Y	Y	Y	Var char2
Date	The date for which the data was contributed.	Y	Y	Y	Y	Date
DocClause	The documentation clause: MM=Modified Modified Restructuring MR=Modified Restructuring CR=Old Restructuring XR=No Restructuring	Y	Y	Y	Y	Char 2
ImpliedRating	Calculated on a weekly basis by comparing the issuer's 5Y senior standard trading convention spread to the 5Y spreads of our sector curves and applying the rating of the logarithmically nearest rating curve specific to that sector.	Y	Y	Y	Y	Char 3
IsBackwardation	A Y/N flag to indicate whether a customer's docclause is backwardated. The docclause that is not conventionally traded			Y		Char1

	for that currency is rejected.					
IsOutlier	Y/N flag to indicate that this point failed the outlier t-test.			Y		Char 1
IsStale	Y/N flag to indicate that this point is stale.			Y		Char 1
IQRange	Interquartile range of all contributions that are considered for the outlier test (calculated for par values). Blank for recoveries.			Y		
Maturity	The maturity of this point in a relative date format, e.g. 1y, 5y, etc.			Y		Char 3
Missing5Y	A Y/N flag indicating if the contributed curve is missing the 5y tenor			Y		
MissingRecovery	A Y/N flag indicating if the contributed curve is missing a recovery rate			Y		
ParSpread	The Par Spread equivalent of the contributed level			Y		
ParSpread6m	The Par Spread equivalent of the contributed value at the 6m tenor		Y			
ParSpread1y	The Par Spread equivalent of the contributed value at the 1y tenor		Y			
ParSpread2y	The Par Spread equivalent of the contributed value at the 2y tenor		Y			
ParSpread3y	The Par Spread equivalent of the contributed value at the 3y tenor		Y			
ParSpread4y	The Par Spread equivalent of the contributed value at the 4y tenor		Y			
ParSpread5y	The Par Spread equivalent of the contributed value at the 5y tenor		Y			
ParSpread7y	The Par Spread equivalent of the contributed value at the 7y tenor		Y			
ParSpread10y	The Par Spread equivalent of the contributed value at the 10y tenor		Y			
ParSpread15y	The Par Spread equivalent of the contributed value at the 15y tenor		Y			
ParSpread20y	The Par Spread equivalent of the contributed value at the 20y tenor		Y			
ParSpread30y	The Par Spread equivalent of the contributed value at the 30y tenor		Y			
ParSpreadXcomposite	The composite par spread without your contribution being used in the calculation			Y		
PointRating	A quantitative measure of the quality of our CDS data at this point.			Y		Varchar2

Primary Coupon	The coupon used when converting Par Spreads to Upfronts in the data cleaning process		Y			
Quality	The overall data quality flag. 'P' for passed and 'F' for failed.			Y		Char 1
Rating6M	A quantitative measure of the quality of our data for a 6M CDS.	Y			Y	Var char2
Rating1Y	A quantitative measure of the quality of our data for a 1Y CDS.	Y			Y	Var char2
Rating2Y	A quantitative measure of the quality of our data for a 2Y CDS.	Y			Y	Var char2
Rating3Y	A quantitative measure of the quality of our data for a 3Y CDS.	Y			Y	Var char2
Rating4Y	A quantitative measure of the quality of our data for a 4Y CDS.	Y			Y	Var char2
Rating5Y	A quantitative measure of the quality of our data for a 5Y CDS.	Y			Y	Var char2
Rating7Y	A quantitative measure of the quality of our data for a 7Y CDS.	Y			Y	Var char2
Rating10Y	A quantitative measure of the quality of our data for a 10Y CDS.	Y			Y	Var char2
Rating15Y	A quantitative measure of the quality of our data for a 15Y CDS.	Y			Y	Var char2
Rating20Y	A quantitative measure of the quality of our data for a 20Y CDS.	Y			Y	Var char2
Rating30Y	A quantitative measure of the quality of our data for a 30Y CDS.	Y			Y	Var char2
Real Recovery	Composite recovery level.				Y	
Recovery	The recovery rate.	Y	Y		Y	Float
RecoveryXcomposite	The composite recovery level without your contribution being used in the calculation			Y		
RedCode	The six-digit RED code.	Y	Y	Y	Y	Char 6
Region	The region of the reference entity.	Y	Y	Y	Y	Char 20
Running Coupon	The coupon to be paid annually until maturity in addition to the 'Upfront' payment.				Y	

Sector (ICB)	The ICB industry sector of the reference entity.	Y	Y	Y	Y	Char 50
ShortName	The abbreviated short name for the reference entity.	Y	Y	Y	Y	Char 100
Spread6m	The spread for this point on the curve.	Y				Float
Spread1y		Y				Float
Spread2y		Y				Float
Spread3y		Y				Float
Spread4y		Y				Float
Spread5y		Y				Float
Spread7y		Y				Float
Spread10y		Y				Float
Spread15y		Y				Float
Spread20y		Y				Float
Spread30y		Y				Float
SpreadOrRate	The credit spread or recovery rate.			Y		Float
StaleDays	The number of days for which this spread has not changed.			Y		Int
Standard Curve	Indicates via 'Y' or 'N whether it is a default curve (i.e. default currency and doc clause combination).		Y	Y		
StdDev	Standard deviation of all contributions that are considered for the outlier test (calculated for par values). Blank for recoveries.			Y		
Ticker	The ticker for the reference entity.	Y	Y	Y	Y	Char 100
Tier	The seniority of the instrument. See table of seniority levels.	Y	Y	Y	Y	Char 8
Ttest	The outlier t-test score for this point.			Y		Float
Type	The type of the instrument. Used to indicate whether this is a recovery or a default swap.			Y		Char 20
Upfront	The upfront equivalent of the contributed level			Y		
Upfront6m	The upfront value for the corresponding tenor		Y		Y	Float
Upfront1y			Y		Y	Float
Upfront2y			Y		Y	Float
Upfront3y			Y		Y	Float
Upfront4y			Y		Y	Float
Upfront5y			Y		Y	Float
Upfront7y			Y		Y	Float

Upfront10y			Y		Y	Float
Upfront15y			Y		Y	Float
Upfront20y			Y		Y	Float
Upfront30y			Y		Y	Float

CDS Theoretical Report Columns

Field	Description	Type
AvRating	The average of the Moody's and S&P ratings adjusted to the seniority of the instrument and rounded to not include the '+' and '–' levels.	VarChar
Coeff1	Coefficient used in a given model to generate this curve.	Float
Coeff2	Coefficient used in a given model to generate this curve.	Float
Coeff3	Coefficient used in a given model to generate this curve.	Float
Coeff4	Coefficient used in a given model to generate this curve.	Float
Coeff5	Coefficient used in a given model to generate this curve.	Float
Coeff6	Coefficient used in a given model to generate this curve.	Float
Coeff7	Coefficient used in a given model to generate this curve.	Float
Coeff8	Coefficient used in a given model to generate this curve.	Float
Coeff9	Coefficient used in a given model to generate this curve.	Float
Coeff10	Coefficient used in a given model to generate this curve.	Float
CompositeRecovery	The recovery rate.	Float
Contributor	Indicates whether the contribution was your own with 'Yours'.	VarChar
Country	The country of the reference entity.	VarChar
Currency	The currency of the instrument.	VarChar
Date	The date for which the data was contributed.	Date
DocClause	The documentation clause.	VarChar
ImpliedRating	Implied Ratings are calculated on a weekly basis by comparing the issuer's 5Y senior standard trading convention spread to the 5Y spreads of our sector curves and applying the rating of the logarithmically nearest rating curve specific to that sector.	VarChar
Model	The model used to calculate this theoretical curve	VarChar
RedCode	The six-digit RED code.	VarChar
Region	The region of the reference entity.	VarChar

Field	Description	Type
Sector	The ICB industry sector of the reference entity.	VarChar
ShortName	The abbreviated short name for the reference entity.	VarChar
Spread6M	The spread for this point on the curve.	Float
Spread1Y		Float
Spread2Y		Float
Spread3Y		Float
Spread4Y		Float
Spread5Y		Float
Spread7Y		Float
Spread10Y		Float
Spread15Y		Float
Spread20Y		Float
Spread30Y		Float
Ticker	The ticker for the reference entity.	VarChar
Tier	The seniority of the instrument.	VarChar

LCDS Report Columns

Field	Description	LCDS Composite	Type
Cancellability	The cancellability of the LCDS contract	Y	VarChar9
Ticker	The ticker for the reference entity.	Y	VarChar30
ShortName	The abbreviated short name for the reference entity.	Y	VarChar100
RedCode	The six-digit RED code.	Y	VarChar6
Tier	The seniority of the instrument. See table of seniority levels on page 75.	Y	VarChar8
CompositeCurveRating	A quantitative measure of the quality of our CDS data, averaged over the points on this curve.		VarChar3
Date	The date for which the data was contributed.	Y	Date
Ccy	The currency of the instrument.	Y	Char3
DocClause	The documentation clause:	Y	Char2
Spread6m	The spread for this point on the curve.	Y	Float
Spread1y		Y	Float
Spread2y		Y	Float
Spread3y		Y	Float
Spread4y		Y	Float
Spread5y		Y	Float
Spread7y		Y	Float

Spread10y		Y	Float
Spread15y		Y	Float
Spread20y		Y	Float
Spread30y		Y	Float
ImpliedRating	Calculated on a weekly basis by comparing the issuer's 5Y senior standard trading convention spread to the 5Y spreads of our sector curves and applying the rating of the logarithmically nearest rating curve specific to that sector.	Y	Char3
CompositeDepth6m	The number of distinct contributors to the composite point.	Y	Int
CompositeDepth1y		Y	Int
CompositeDepth2y		Y	Int
CompositeDepth3y		Y	Int
CompositeDepth4y		Y	Int
CompositeDepth5y		Y	Int
CompositeDepth7y		Y	Int
CompositeDepth10y		Y	Int
CompositeDepth15y		Y	Int
CompositeDepth20y		Y	Int
CompositeDepth30y		Y	Int
Owner	For Composite Curves, the owner will state “Composite”	Y	VarChar9
Recovery	The recovery rate.	Y	Float
RecoveryDepth	The number of distinct contributors to the recovery rate	Y	Int
Sector (ICB)	The ICB industry sector of the reference entity.	Y	Char50
Region	The region of the reference entity.	Y	Char20
AvRating	The average of Moody’s and S&P ratings adjusted to instrument’s seniority and rounded to not include ‘+’ and ‘–‘ levels.	Y	Char3

Bond Report Columns

The following table describes the columns for the Bond reports.

Field	Description	Bond Composites	Bond Contributions	Bond DataQuality	Type
AvRating	The average of the Moody’s and S&P ratings adjusted to the seniority of the instrument and rounded to not include the ‘+’ and ‘–‘ levels.	Y	Y	Y	Char3

Field	Description	Bond Composites	Bond Contributions	Bond DataQuality	Type
BenchmarkName	The underlying benchmark bond.	Y			Varchar2
BenchmarkSpread	The yield spread of the underlying benchmark yield.	Y			Float
BenchmarkYield	The yield of the benchmark.	Y			Float
CarryFwd	Indicates the number of days a contribution has been carried forward. A price is carried forward by adjusting for the average change in all good points from one day to the next.		Y	Y	Int
Ccy	The currency of the instrument.	Y	Y	Y	Char3
CompositeCDSImpliedSpread	The spread generated by evaluating the credit curve for the issuing entity at bond maturity.	Y	Y	Y	Float
CompositeDefaultProbSpread	The spread that must be applied to the issuing entity's credit curve to make a default probability PV match the dirty price.	Y			Float
CompositePrice	The average price of all good contributions.	Y	Y		Float
CompositePriceAsk	The average ask price of all good contributions.	Y	Y		Float
CompositePriceBid	The average bid price of all good contributions.	Y	Y		Float
CompositePriceRating	A quantitative measure of the quality of our bond data.	Y		Y	Varchar2
CompositeUsdParAswSpread	The asset swap spread calculated from the CompositePrice.	Y	Y		Float
Contributor	Indicates whether the contribution was your own with ‘Yours’; otherwise, this field is blank.	Y	Y		Char10
Country	The country of the issuing organization.	Y	Y	Y	Varchar2
Coupon	The bond’s coupon. For FRNs, the coupon is 0.	Y	Y	Y	Float
Cusip	The bond’s CUSIP.	Y	Y	Y	Char9
Date	The date for which the data was contributed.	Y	Y	Y	Date
Depth	The number of distinct contributors.		Y	Y	Int

Field	Description	Bond Composites	Bond Contributions	Bond DataQuality	Type
Distance	The percentage distance from the mean of a contribution.			Y	Float
ImpliedRating	Calculated on a weekly basis by comparing the issuer's 5Y senior standard trading convention spread to the 5Y spreads of our sector curves and applying the rating of the logarithmically nearest rating curve specific to that sector.	Y	Y	Y	Char3
Instrument	The instrument name.	Y	Y	Y	Char100
Isin	The bond's ISIN.	Y	Y	Y	Char12
IsOutlier	Y/N flag to indicate that this point failed the outlier t-test.			Y	Char1
IsStale	Y/N flag to indicate that this point is stale.			Y	Char1
Maturity	The bond's maturity date.	Y	Y	Y	Date
Price	The price.		Y	Y	Float
PriceAsk	The ask price.		Y	Y	Float
PriceBid	The bid price.		Y	Y	Float
Quality	The overall data quality flag. 'P' for passed and 'F' for failed.			Y	Char1
Region	The region of the reference entity.	Y	Y	Y	Char20
Sector	The ICB industry sector of the reference entity.	Y	Y	Y	Char50
ShortName	The abbreviated short name for the reference entity.	Y	Y	Y	Char100
SnapTime	The time of day (London local time) at which the bond price was snapped.		Y	Y	Time
StaleDays	The number of days for which this bond price has not changed.			Y	Int
Ticker	The ticker of the bond issuer.	Y	Y	Y	Char100
Tier	The seniority of the instrument. See the table of seniority levels.	Y	Y	Y	Char8
Ttest	The outlier t-test score for this point.			Y	Float
UsdParAswSpread	The dollar par asset swap spread calculated from the Price.		Y	Y	Float

Field	Description	Bond Composites	Bond Contributions	Bond DataQuality	Type
XCompositePrice	The composite price excluding your own.			Y	Float
XCompositePriceAsk	The composite ask price excluding your own.			Y	Float
XCompositePriceBid	The composite bid price excluding your own.			Y	Float
XCompositeUsdParAswSpread	The composite dollar par asset swap spread excluding your own.			Y	Float

Bond Theoretical Report Columns

Field	Description	Type
AvRating	The average of the Moody's and S&P ratings adjusted to the seniority of the instrument and rounded to not include the '+' and '-' levels.	VarChar
CCY	The currency for the bond.	VarChar
Country	The country of the reference entity.	VarChar
Coupon	The bond's coupon. For FRNs, the coupon is zero (0).	Float
Cusip	The unique CUSIP for this bond.	VarChar
Date	The date for which the data was contributed.	Date
ImpliedRating	The Implied Ratings are calculated on a weekly basis by comparing the issuer's 5Y senior standard trading convention spread to the 5Y spreads of our sector curves and applying the rating of the logarithmically nearest rating curve specific to that sector.	VarChar
Instrument	The common instrument name for the bond.	VarChar
Isin	The unique ISIN for this bond.	VarChar
Maturity	The final maturity date for the bond.	Date
Model	The model used to calculate this theoretical. This will always be 'CDSImplied'.	VarChar
ModelCDSSpread	The spread from the theoretical calculation.	Float
ModelPrice	The price from the theoretical calculation.	Float
Region	The region of the reference entity.	VarChar
Sector	The industry sector of the reference entity.	VarChar
ShortName	The abbreviated short name for the reference entity.	VarChar
Ticker	The ticker of the bond issuer.	VarChar
Tier	The tier or seniority of the bond.	VarChar

Sector Report Columns

Field	Description	Type
Type	Indicates the type of data contained within a report (dependent on the Type and Report chosen). Values are: - Sector -- when Type "Levels" and Report "Par" is chosen - sectorZero -- when Type "Levels" and Report "Zero" is chosen Note that the value will be common to all entries within a given report.	Char(12)

Field	Description	Type
Date	The date of the sector curve.	Date
IndustryName	The ICB industry classification. Note: as ICB does not research sovereign entities Markit uses an additional classification of “Government”. Values are: Basic Materials Consumer Goods Consumer Services Financials Government Health Care Industrials Oil & Gas Technology Telecommunications Utilities <blank> Blank entry indicates a cross-industry rating-only (or generic) curve.	Char(50)
Interpolated	Flags whether the curve is interpolated (Y= interpolated; N = not interpolated).	Char(1)
PrevDate	For cases where the report type “Changes” is used this shows the previous business day. Note: this is only relevant for the older (v5) report.	Date
Rating	Senior unsecured rating. Values are: AAA AA A BBB BB B CCC	Char(3)
Recovery	The mean recovery rate of the constituent curves.	Float
Spread6M	The mean spread across the constituent entities for the given point on the curve.	Float
Spread1Y		Float
Spread2Y		Float
Spread3Y		Float
Spread4Y		Float
Spread5Y		Float
Spread7Y		Float
Spread10Y		Float
Spread15Y		Float

Field	Description	Type
Spread20Y		Float
Spread30Y		Float

Credit Index Report Columns

Field	Description	Type
Composite Price	The top and tailed (quartiles) and then averaged index price.	Float(126)
Composite Spread	The top and tailed (quartiles) and then averaged index spread.	Float(126)
Date	The date of pricing information.	Date
Depth	The number of good contributions used to calculate the composite for this instrument.	Number(3)
Heat	The heat calculated as today's absolute change in spread divided by the absolute average daily change over the month excluding today's change.	Float(126)
Index ID	The Markit index ID.	Varchar2(100)
Maturity	The index maturity date.	Date
Model Price	The theoretical price (based on components).	Float(126)
Model Spread	The theoretical spread (based on components).	Float(126)
Name	The index name.	Char(100)
On The Run	Y/N flag to indicate if this is the latest series and version of the index.	Char(1)
RED Code	The RED Code of the index.	Varchar2(9)
Series	The index series number.	Number
Term	The index period.	Varchar2(4)
Version	The index version number.	Number

Credit Index Annex Report Columns

Please refer to XML Technical Documentation for Markit RED for details of the Credit Index Annex Report. You may also request the XSD for this report from Markit (see the Contact Us section52 of this guide).

ABS Report Columns

Field	Description	ABS Composite	ABS Contributions	ABS Data Quality
Avlife	The average life of the instrument before prepayment.	Y	Y	Y
AvRating	The average of the Moody's and S&P ratings adjusted to the seniority of the instrument and rounded to not include the '+' and '-' levels.	Y	Y	Y
Benchmark	For FRNs, this is the fixing rate. Null for fixed coupons.	Y	Y	Y
Ccy	The ISO currency of the instrument.	Y	Y	Y
CompositePriceBid/Mid/Ask	The average ask price of all good contributions.	Y		
CompositeSpreadBid/Mid/Ask	The average ask spread of all good contributions.	Y		

Field	Description	ABS Composite	ABS Contributions	ABS Data Quality
Contributor	Indicates whether the contribution was your own with 'Yours', and whether it is a composite with 'Composite'.	Y	Y	
Coupon Type	The ABS coupon type, which is Fixed or a Floating Rating Note (FRN).	Y	Y	Y
Coupon	The ABS coupon.	Y	Y	Y
Cusip	The ABS CUSIP.	Y	Y	Y
Date	The date for which the data was contributed.	Y	Y	Y
Depth	The number of distinct contributors for this ABS instrument.	Y	Y	Y
Distance	The percentage distance between your submitted point and the mean of all the others.			Y
DealTicker	The ticker of the ABS issuer.	Y	Y	Y
DealName	The abbreviated short name for the ABS deal.	Y	Y	Y
Instrument	The instrument name.	Y	Y	Y
Isin	The ABS ISIN.	Y	Y	Y
IsOutlier	Y/N flag to indicate that this point failed the outlier t-test.			Y
Maturity	The ABS contractual maturity date.	Y	Y	Y
PriceAsk	The ask price.		Y	Y
PriceBid	The bid price.		Y	Y
PriceMid	The mid price.		Y	Y
Prepay	The prepayment rate.		Y	Y
PrepayType	The prepayment rate type.		Y	Y
Quality	The overall data quality flag. 'P' for passed and 'F' for failed.			Y
Region	The region of the reference entity.	Y	Y	Y
SnapTime	The time of day (London local time) at which the bond price was snapped.		Y	Y
SpreadAsk	The ask spread.		Y	Y
SpreadBid	The bid spread.		Y	Y
SpreadMid	The mid spread.		Y	Y
Ttest	The outlier t-test score for this point.			Y

Sameday CDS Report Columns

Field	Description	Type
AvRating	The average of the Moody's and S&P ratings adjusted to the seniority of the instrument and rounded to not include the '+' and '–' levels.	VarChar

Field	Description	Type
Ccy	The ISO currency of the instrument.	VarChar
Country	The country of the reference entity.	VarChar
DataRating	A quantitative measure of the quality of Markit data.	VarChar
Date	The date for which the data was contributed.	Date
DocClause	The documentation clause.	VarChar
ImpliedRating	Calculated on a weekly basis by comparing the issuer's 5Y senior standard trading convention spread to the 5Y spreads of our sector curves and applying the rating of the logarithmically nearest rating curve specific to that sector.	VarChar
RedCode	The six-digit RED code.	VarChar
Region	The region of the reference entity.	VarChar
ShortName	The abbreviated short name for the reference entity.	VarChar
Spread6M	The spread for this point on the curve.	Float
Spread1Y		Float
Spread2Y		Float
Spread3Y		Float
Spread4Y		Float
Spread5Y		Float
Spread7Y		Float
Spread10Y		Float
Spread15Y		Float
Spread20Y		Float
Spread30Y		Float
Ticker	The ticker for the reference entity.	VarChar
Tier	The seniority of the instrument. See table of seniority levels.	VarChar
TimeZone	The time zone indicating the market close. Values are: L = London; N = New York; T= Tokyo; E = Europe	VarChar

Appendix F – Table of Composite Fallback Levels

Flag	Contributors	Boundary
Currency Grouping (CcyGrp)	3 or more contributions from anyone	Three distinct groups: AUD; JPY; Everything else (USD, EUR, GBP, etc). For the same documentation clause (DocClause).

Documentation clause Adjusted
(DocAdj)
	3 or more contributions from anyone	For currencies that share the group (as per above)
		An adjustment is made between different documentation clauses by applying the following ratios (which are expressed as the factor used to adjust back to CR):
		• CR: 1.0
		• MM: 0.96
		• MR: 0.95
		• XR: 0.935*
		• Japanese XR: 0.75
		The algorithm works by first adjusting all spreads to CR with the following formula:
		`CR_Avg=sum(submitted_spread / ratio) / number_of_spreads`
		This amount is then adjusted to the required documentation clause average by multiplying by the ratio, e.g.
		`MM_Avg=CR_Avg * 0.96`

EntityTier	3 or more contributions from anyone	By entity and tier, i.e. averaging across currency and documentation clause boundary.
Thin	2 contributions from anyone	By entity and tier.

Appendix G – Table of Seniority Levels

Seniority or Tier Abbreviation	Description
JRSUBUT2	Junior Subordinated or Upper Tier 2 Debt (Banks)
LIEN1	First Lien – Secured Debt with a First Lien on a pool of assets. (Introduced in August 2006)
LIEN2	Second Lien – Secured Debt with a Second Lien on a pool of assets. (Introduced in August 2006)
LIEN3	Third Lien – Secured Debt with a Third Lien on a pool of assets. (Introduced in August 2006)
MEZZ	Mezzanine – Contractually or Structurally Subordinated, Unsecured Debt falling between senior debt and equity. Commonly used in leveraged buyouts or by middle-market companies. (Introduced in August 2006)
PREFT1	Preference Shares, or Tier 1 Capital (Banks)
SECDOM	Secured Debt (Corporate/Financial) or Domestic Currency Sovereign Debt (Government)
SNRFOR	Senior Unsecured Debt (Corporate/Financial), Foreign Currency Sovereign Debt (Government)
SUBLT2	Subordinated or Lower Tier 2 Debt (Banks)

Appendix H – Table of Document Clauses

Doc Clause Abbreviation	Description
CR	Cum-Restructuring or Old Restructuring
MR	Modified Restructuring
MM	Mod-Mod Restructuring (Introduced in June 2003)
XR	Ex-Restructuring