JSON Market Reform Contract (JMRC) Specification

Version 0.5
Published 21/10/2021

WHITESPACE
Sequel
A Verisk Business

Purpose

To define the **JMRC** (JSON Market Reform Contract) standard developed by Whitespace and subsequently donated to ACORD for industry-wide use.

Background

Digital Placing Standard

At Whitespace we started work on digital contracts for the London Insurance market (and beyond) in 2014. We wanted to find a way to create, negotiate, trade, process and endorse fully digital contracts, in order to help the market to realise the savings a digital transformation will enable.

Traditionally, agreeing new data standards for the London Market is slow and the resultant standards are often complex and expensive to adopt. Whitespace took the decision to use an **existing business standard**, the Market Reform Contract (MRC), and represent that using an **existing global data standard** (JSON). The aim was to effectively merge these existing standards in order to create a common standard that can easily be understood by both business people and IT teams.

The **JMRC** (JSON Market Reform Contract) is the resulting digital placing standard that Whitespace has now donated to ACORD, the standards-setting body for the global insurance industry, and which will be incorporated into ACORD's industry-owned Global Reinsurance and Large Commercial Data Standards, which are widely used through the global insurance value chain.

It should be noted that, whilst the MRC is the starting place for this **JMRC** standard, other contract structures can already be supported, as are multiple presentation layouts. Standard MRC and non-standard headings alike are recognised and the associated data is stored in the digital contract – refer to the **MRCHeading** definition in **Fig 2** below.

Whitespace is fully committed to the continued development and improvement of the **JMRC**. Our future direction will be to ensure that a wider variety of contract types and further presentation layouts can be fully accommodated. Expect updated versions of **JMRC** standard in the upcoming months, including a possible rename in order to better reflect our wider market vision.

Semi-structured AND Structured Data

The **JMRC** standard provides a high level of flexibility, storing the entirety of the contract data in a semi-structured manner under standard or custom headings.

With the latest release of the Whitespace Platform (version 2.0) we see the introduction of **defined data tagging**. Without changing the presentation of the contract to the end client or affecting the semi-structured contract data already available, the **defined data tagging** approach extends the existing **placeholder**

concept to give precise definition to key data values within the contract data which would otherwise require some level of interpretation logic to identify.

Advances in technology make it possible to analyse semi-structured data which, in tandem with the **tagged defined data** values, gives an organisation complete access to their contract wording data along with certainty as to those key data values. That certainty is shared by and common to all participants on a contract. The traditional importance of having all of the data organised in exactly the same structured schema is met, without losing the supporting contextual data and without risking duplication or misinterpretation of information.

The **JMRC** includes certain ‘Core’ objects and name-value pairs which are mandatory to the integrity of the MRC record and which are documented below. The ‘Core’ objects also include a small handful of additional name-value pairs that have direct business impact or relevance but which are not specifically a part of the MRC itself, such as **status** (which shows whether the risk is a Draft, a Firm Order, or if it has been Signed already) and **index** (which is necessary to maintain the order of the data as it is presented on an individual MRC).

Furthermore, the **JMRC** is flexible enough to support ‘Extensions’ which can be included if required for other purposes such as system specific controls. At Whitespace, for example, we have used some ‘Extensions’ that are specific to the Whitespace Platform and are used by the system when placing and progressing the risk. Such ‘Extensions’ are not documented here.

The JMRC Described

This **JMRC** specification covers three different types of JSON document: the Placing document (RWPlacing), the Written Line document (**RWWrittenLineSet**) and the Signed Line document (**RWSignedLineSet**). All three types contain common objects, with some slight variations. (This document highlights the variations. If no mention is made it can be assumed that they are common to all three.)

The **JMRC** comprises a number of objects the principal of which is the Core **MRCContract** object which contains an array of **lineItems** (or name-value pairs) holding the actual text of the contract as data. In addition to the **MRCContract** object there are a number of other objects which hold information relating to the creation of the JSON document itself, including summary and control data (see table Fig 1 below for a description of each of the JMRC objects).

The **JMRC** does not impose any data standards beyond those determined by the MRC.

Fig 1

Values	Description
MRCContract	Holds the complete contractual text as an array of **lineItems** (see table Fig 2 below).
control	Holds summary data about the contract, such as its status, the placing broker and the lead underwriter (see table Fig 3 below). **Variation**: the control object does not form part of the Written Line (**RWWrittenLineSet**) or the Signed Line (**RWSignedLineSet**) documents
createdAt	Holds the date on which the record was first created
type	Identifies which type of JSON document it is: - **RWPlacing** - **RWWrittenLineSet** - **RWSignedLineSet**
updatedAt	Holds the date on which the record was last updated

RWPlacing

The **RWPlacing** document contains the complete text of the contract, including Headings and Section Headings, under the **MRCContract** section heading along with supporting informational data as described above.

Example JSON:

The following shows the outline of the previously defined objects contained in the **JMRC RWPlacing** document

```
{
  "MRCContract": {
    "lineItems": [ "{{LineItem}}" ]
  },
  "control": "{{JMRC-Control}}",
  "createdAt": "2019-08-19 07:35:09",
  "type": "RWPlacing",
  "updatedAt": "2019-08-19 07:35:09"
}
```

MRCContract

The **MRCContract** object comprises an array of **lineItems** (name-value pairs or nested arrays) which contain, order and define the actual contract data (see table Fig 2 below for a description of each of the **lineItems** that make up the **MRCContract** object).

Fig 2

Value	Description
elements	An array of **lineItems** which preserve all text on the actual contract document. Only text strings, bullet or numbered lists, and tables are supported currently. **JSON example**: simple text string • The text to be displayed for each specific paragraph within a **lineItem** is defined • The order of the of that **text** within the **lineItem** is defined by the **index** (i.e. for paragraphs of **text**) • To support limited rich text formatting in systems or applications capable of rendering them, **html** holds a

Value	Description

	duplicate of the **text** value but with **html** tags to support bold, italic and underline formatting where such formatting has been applied to any of the **text**. If no formatting has been applied the **html** name-value pair will not be present.
	
	Note: cascading of styles is not supported
	
	`"elements": [ { "index": 0, "text": "B0999ABC123456789" "html": "<b>B0999ABC123456789</b>" } ],`
	
	JSON example: 2 point bullet list
	
	• The value or character to be used for the bullet is defined (**bulletValue**)
	• The **indent** from the left hand margin (where -1 is fully left aligned and 0 is indented once) is defined
	• The order of the of the **text** within the list is defined by the **index**
	• The **text** to display for each point in the list is defined
	
	`"elements": [ { "bulletValue": "-", "indent": 0, "index": 0, "text": "Inclusive fares;" }, { "bulletValue": "-", "indent": 0, "index": 1, "text": "Free food and drinks on all long haul flights;" }, ],`

Value	Description

	JSON example: 2 point numbered list
	• As defined in the bullet list example above, the numeric identifier (i.e. “1.” Or “1)”) to be used for each item in the list is defined (**bulletValue**)
	• The **indent** from the left hand margin (where -1 is fully left aligned and 0 is indented once) is defined
	• The order of the of the **text** within the list is defined by the **index**
	• The **text** to display for each point in the list is defined
	
	"elements": [
	{
	"bulletValue": "(1)",
	"indent": 0,
	"index": 0,
	"text": "payment by the Reinsurer has not already been made or credited to the Reinsured;"
	},
	{
	"bulletValue": "(2)",
	"indent": 0,
	"index": 1,
	"text": "the Reinsured becomes Insolvent (as defined below);"
	},
	],
	
	JSON example: 2 row table with a header row
	• Each of the **rows** is defined, including any headings, with a specified **height** and a list of values for each column
	• Each column is defined with specified **widths** and the **columnSpecs** sets out the column alignment
	
	"elements": [
	{
	"index": 0,
	"table": {
	"rows": [
	{
	"height": 21,
	"values": [
	"Reg",
	"Aircraft Type",
	"Config",
	"Delivered",

Value	Description


```
                "No.",
                "Age"
            ]
        },
        {
            "height": 21,
            "values": [
                "C-FYJI",
                "Airbus A319-100",
                "C14Y106",
                "May-97",
                "258",
                "21.0 Years"
            ]
        },
        {
            "height": 21,
            "values": [
                "C-FYKC",
                "Airbus A319-100",
                "C14Y106",
                "Jun-97",
                "222",
                "20.9 Years"
            ]
        }
    ],
    "rowsSpecs": [],
    "widths": [
        68,
        101,
        71,
        75,
        41,
        60
    ],
    "columnsSpecs": [
        {
            "alignment": "centre",
            "x": 0
        },
        {
            "alignment": "left",
            "x": 0
        },
        {
            "alignment": "left",
            "x": 0
        },
        {
            "alignment": "left",
            "x": 0
        },
```

Value	Description

	{
	"alignment": "right",
	"x": 0
	},
	{
	"alignment": "left",
	"x": 0
	}
	]
	}
	}
	],

originalHeading	The Heading as displayed on the original contract and the Heading displayed in any visual representation of the **JMRC**, i.e. within the Whitespace Platform or once converted to a .pdf or Word document. See **Worked Example 1** below for further clarification
mrcHeading	The standard MRC Heading to which the **originalHeading** of the **lineItem** is mapped. For example, where the **originalHeading** is “Territory” this would be mapped to the **mrcHeading** “Situation”. The **mrcHeading** may also contain the value “CustomMRCHeading” where an **originalHeading** value exists which has not been mapped to any of the standard MRC Headings. This ensures that all data in the original contract can be stored in the **JMRC**, even if it is under a non-standard MRC Heading. See **Worked Example 1** below for further clarification.
nameVariation	Holds one of a pre-defined set of alternative headings that map to the **mrcHeading**. For example, where the **originalHeading** is “RATE:” this would be mapped to the **mrcHeading** “Premium” as “Rate” is a recorded **nameVariation** for “Premium”. Refer to the MRC Data Dictionary or to the SDC Data Extraction and Mapping spreadsheet linked in the **Reference**

Value	Description

	Material section at the end of this document for details of the possible values.
	
	{
	"elements": [
	{
	"index": 0,
	"text": "Premium Rate [[Percentage1]]"
	}
	],
	"index": 17,
	"mrcHeading": "Premium",
	"mrcSection": "RiskDetails",
	"nameVariation": "Rate",
	"originalHeading": "RATE:",
	"originalSection": "",
	"placeholders": [
	{
	"name": "Percentage1",
	"phType": "Percentage",
	"value": "1.5 %"
	}
	],
	"subItems": []
	},

originalSection	**originalSection** will either: hold the value “self” indicating that the lineItem in question contains the Section Heading itself or be blank Where the **originalSection** value is “self”, the **originalHeading** will contain the Section Heading as displayed on the contract. See **Worked Example 2** below for further clarification
mrcSection	The standard MRC Section Heading under which each **lineItem** should be displayed. This value contains no spaces (i.e. “RiskDetails” rather than “Risk Details”) See **Worked Example 2** below for further clarification

Value	Description
index	The **index** value is used to determine the order or sequence of each ***lineItem*** within the **MRCContract** as a whole (in addition to the **index** ordering within a specific **lineItem**)

placeholders	**placeholders** are applied to the following data values identified in the **text**:
	• Numbers
	• percentages
	• dates
	• time durations
	• email addresses
	• clauses
	Any such values will be replaced by placeholder **name** values within the text **lineItem**. Each placeholder **name** is then defined within the **placeholders** array. Each placeholder **name** is restated along with the **phType** defining the applicable data type (i.e. Date), and the **value** from the **text** that has been replaced by the placeholder **name**.
	<pre>{
	"elements": [
	{
	"index": 0,
	"text": "[[Date1]]"
	}
	],
	"index": 36,
	"mrcHeading": "SettlementDueDate",
	"mrcSection": "SubscriptionAgreement",
	"nameVariation": "SettlementDueDate",
	"originalHeading": "SETTLEMENT DUE DATE:",
	"originalSection": "Subscription Agreement",
	"placeholders": [
	{
	"name": "Date1",
	"phType": "Date",
	"value": "1 April 2019"
	"originalValue": "1 April 2019"
	}
	],
	"subItems": []
	},</pre>
	The **defined data** release introduces an extension to the **placeholder** functionality and allows for any text value or values within a **lineItem** to be **tagged**, so long as the **tagged**

Value	Description

	value(s) meet the specific **tag**'s validation requirements. The tags are not limited to the standard **placeholder** types.
	
	Any values which have been **tagged** will have a **phType** of "DefinedData". In addition, the **name** value in the resulting placeholder will be in the format "DefinedData[n]". The rest of the placeholder information is the same as a standard **placeholder**, except for the addition of the **tags** array.
	
	The **tags** array will contain at least two values, a **page** identifier (see **tagPages** below) and the defining **tag** which has been applied to the **value**.
	
	It is possible for more than one **tag** to be applied to a single **value**. If this is the case, all applicable **tags** will be listed in the **tags** array.
	
	The **tags** available per **mrcHeading** and any associated validations/look up lists of values are defined in the **RWDefinedData** document.
	
	"elements": [
	{
	"index": 0,
	"text": "From [[DefinedData0]] to [[DefinedData1]]"
	}
	],
	"placeholders": [
	{
	"name": "DefinedData0",
	"phType": "DefinedData",
	"value": "1 August 2021",
	"tags": [
	"Page:D36C8259-08A9-4220…1993",
	"Inception_Date"
	],
	"originalValue": "1 August 2021"
	},
	{
	"name": "DefinedData1",
	"phType": "DefinedData",
	"value": "31 August 2022",
	"tags": [
	"Page:D36C8259-08A9-4220…1993",
	"Expiry_Date"
	],
	"originalValue": "31 August 2022"
	}
	]

Value	Description
tagPages	**Tags** within a **lineItem** may be grouped together to form a collection of specifically related values. For example, where more than one Premium amount is tagged within the Premium **lineItem**, information defining the first amount can be grouped so that it is distinct from information defining the second amount, and so on, building out a transactional view of the data. **tagPages** defines the groups that have been created per **lineItem.** • **pageID** is system generated and is assigned to each “DefinedData” **placeholder** value within the group • **title** is an optional value entered by the user to describe the group (it is envisaged that this will become selectable from a list of values in a future release). • **sectionIDs** can be associated with a specific **pageID**, if sections have been set up for a specific contract There is always at least one group and therefore at least one **pageID** defined under **tagPages**. See **Worked Example 3** below for further clarification `"tagPages": [ { "sectionIDs": [], "title": "Contract Premium", "pageID": "D36C8259-08A9-4220…1993" }, { "sectionIDs": [], "title": "TRIA Premium", "pageID": "69017507-6AB5-4800…1E4C" } ]`
locked	The Broker is able to choose to lock a **lineItem** before showing a contract to Carriers as a request to quote. This restricts what a Carrier is able to edit/update when preparing their quote. Note that values which have DefinedData **tags** may be updated even if the **lineItem** is locked. `"locked": true,`

Worked Example 1 - MRC Headings

Standard MRC Headings - Territory

Referring to the example JSON below, the original contract contains the heading "**TERRITORY**" alongside the text "Worldwide" as the stated location of the risk.

In the **JMRC** document the heading "**TERRITORY**" is retained as the **originalHeading** value and can therefore be displayed exactly as it appeared on the original in any subsequent visual representation of the **JMRC**. It can also be identified by the mapped standard **MRCHeading** when performing any standardised reporting, conversion, or integration of data into other formats or systems ensuring consistency of purpose and definition.

If you refer to the MRC Data Dictionary (which is available to download here: https://www.londonmarketgroup.co.uk/mrc) you will see that this logic has been taken directly from this existing market standard. In the case of this specific example, the *Heading Name* for the “situation, territorial limits or scope, trading warranties or location data” is "Situation", but "Territory" is listed in the Data Dictionary as one of the recognised *Name Variations* for the *Heading Name* "Situation". The **JMRC** has reflected and applied the *Heading Name* and *Name Variations* logic as defined in the MRC Data Dictionary.

Example JSON:


```
{
    "elements": [
        {
            "index": 0,
            "text": "Worldwide"
        }
    ],
    "index": 11,
    "mrcHeading": "Situation",
    "mrcSection": "RiskDetails",
    "nameVariation": "Territory",
    "originalHeading": "TERRITORY:",
    "originalSection": "",
    "placeholders": [],
    "subItems": []
},
```


Custom MRC Headings – Project Site

Where information is present in the original contract under a non-standard MRC Heading, the **JMRC** is able to capture and store that information by treating it as a Custom MRC Heading, thereby ensuring that no data in the original contract is excluded or lost.

Example JSON:


```
{
   "elements": [
      {
         "index": 0,
         "text": "Main Street, Old Town, AB1 2DE"
      }
   ],
   "index": 8,
   "mrcHeading": "CustomMRCHeading",
   "mrcSection": "RiskDetails",
   "nameVariation": "CustomMRCHeading",
   "originalHeading": "PROJECT SITE:",
   "originalSection": "",
   "placeholders": [],
   "subItems": []
},
```


Worked Example 2 - MRC Section Headings

As defined in the MRC Data Dictionary (linked above) there are 6 standard MRC Section Headings:

- **Risk Details**; details of the risk/contract involved, such as insured (or reinsured), type, coverage, conditions etc.
- **Information**: free text additional information.
- **Security Details**: insurer's "stamp" details. These indicate each insurer's share of the risk, their reference/s and any additional conditions applied by them.
- **Subscription Agreement**: in relation to subscription markets, this establishes the rules to be followed for processing and administration of post-placement amendments and transactions.
- **Fiscal and Regulatory**: fiscal and regulatory issues specific to the insurers involved in the risk.
- **Broker Remuneration & Deductions**: Information relevant to brokerage, fees paid to the broker and deductions from premium

If the **lineItem** in the **JMRC** is a line containing the actual Section Heading from the original contract, to be displayed as a **lineItem** in its own right, then:

- The **elements** value will be blank
- The **index** will determine where in the overall ordering of **lineItems** the Section Heading should be displayed
- the **originalHeading** value will state the Section Heading as it appeared in the original contract. In this example the standard Section Heading is "RISK DETAILS"
- the **mrcHeading** will hold a default value of NonMRCHeading
- the **nameVariation** will hold a default value of NonMRCHeading

- the **originalSection** will hold the value “self” indicating that the lineItem is for the Section Heading itself
- the **mrcSection** value will state the standard Section Heading that the lineItem relates to (without any spaces or other punctuation). In this example the standard Section Heading is ''RiskDetails''

Example JSON:

```
{
   "elements": [],
   "index": 0,
   "originalHeading": "RISK DETAILS",
   "mrcHeading": "NonMRCHeading",
   "nameVariation": "NonMRCHeading" ,
   "originalSection": "self",
   "mrcSection": "RiskDetails",
   "placeholders": [],
   "subItems": []
},
```

If the **lineItem** in the **JMRC** is a line containing contract data that should appear under a specific Section Heading on the original MRC document, for example the UMR, then:

- the **mrcSection** value will state the mapped Section Heading that the **lineItem** relates to (without any spaces or other punctuation)
- the **originalSection** value will be blank

Example JSON:

```
{
   "elements": [
      {
         "index": 0,
         "text": " B0999ABC123456789"
      }
   ],
   "index": 1,
   "mrcHeading": "UMR",
   "mrcSection": "RiskDetails",
   "nameVariation": "UMR",
   "originalHeading": "UNIQUE MARKET REFERENCE:",
   "originalSection": "",
   "placeholders": [],
   "subItems": []
},
```

Worked Example 3 – Defined Data tags with pages and sections

Where **tags** have been applied **placeholders** are created or updated with **phType** “DefinedData”. These placeholders also include a **tags** array which will include the

page (or group) associated with each tagged value, and the **tag**(s) that have been applied.

pages are further defined in the **tagPages** object where any user entered page (or group) **title** is recorded along with any section(s) associated.

Example JSON:


```
{
        "mrcSection": "RiskDetails",
        "elements": [
          {
            "index": 0,
            "text": "Hull Section: [[DefinedData0]] ([[Percentage1]]) Annual Premium"
          },
          {
            "index": 1,
            "text": ""
          },
          {
            "index": 2,
            "text": "Liability Section: [[DefinedData1]] ([[Percentage2]]) Annual Premium"
          }
        ],
        "locked": false,
        "originalHeading": "PREMIUM",
        "originalSection": "",
        "nameVariation": "Premium",
        "placeholders": [
          {
            "phType": "Percentage",
            "originalValue": "100%",
            "value": "100%",
            "name": "Percentage1"
          },
          {
            "phType": "Percentage",
            "originalValue": "100%",
            "value": "100%",
            "name": "Percentage2"
          },
          {
            "phType": "DefinedData",
            "originalValue": "USD 5,500,000",
            "value": "USD 5,500,000",
            "tags": [
              "Page:1CF01649-99F6-4E73…F0B2",
              "Premium_Amount"
            ],
            "name": "DefinedData0"
          },
          {
            "phType": "DefinedData",
            "originalValue": "USD 4,500,000",
            "value": "USD 4,500,000",
            "tags": [
              "Page:95738B8F-DD8A-4F8E…4E30",
```

```
                    "Premium_Amount"
                ],
                "name": "DefinedData1"
            }
        ],
        "mrcHeading": "Premium",
        "index": 14,
        "tagPages": [
            {
                "pageID": "1CF01649-99F6-4E73…F0B2",
                "sectionIDs": [
                    "1"
                ],
                "title": "Hull"
            },
            {
                "pageID": "95738B8F-DD8A-4F8E…4E30",
                "sectionIDs": [
                    "2"
                ],
                "title": "Liability"
            }
        ]
    },
```


RWWrittenLineSet

The **RWWrittenLineSet** document contains most of the same objects as the **RWPlacing** document, with the exception of the **control** object which is not repeated.

The **RWWrittenLineSet** specifically contains details of any written line % and stamp values, including underwriter references and risk codes, within the **contents** array (in place of the **elements** array).

Example JSON:


```
"contents": [
 {
  "businessUnit": "I.G.C. Underwriting",
  "conditionsApproved": true,
  "conditionsEnabled": true,
  "impressions": [
   {
    "riskCodes": [
                    {
                        "code": "T,B,W,3T",
                        "index": "0"
                    }
                ],
    "stamp": {
         "bureauMarket": "Lloyd's",
          "bureauMarketCode": "4321",
```

```
            "bureauSubMarket": "",
            "businessUnit": "Lloyd's Syndicate IGC 4321"
            "stampType": "lloyds",
            "uniqueID": "ST579B30FD-1640-45AC…6FCD"
        },
        "uwRefs": [
          "ABC123456A20"
        ],
        "writtenLinePercentageString": "90",
        "yearOfAccount": "2020"
      }
    ],
    "lineConditions": [],
    "sectionIdentifiers": [],
    "stampedAt": "2019-05-28 09:01:51",
    "subjectivities": [],
    "subjectivitiesDeadlineNotRequired": "no deadline / conditions update",
    "toStand": false
  }
],
```


RWSignedLineSet

The **RWSignedLineSet** document is similar to the **RWWrittenLineSet** document. It specifically contains details of the signed line % and stamp values, including underwriter references and risk codes, within the **contents** array (in place of the **elements** array).

Example JSON:


```
"contents": [
  {
    "businessUnit": "I.G.C. Underwriting",
    "conditionsApproved": false,
    "conditionsEnabled": false,
    "impressions": [
      {
        "riskCodes": [
                        {
                            "code": "T,B,W,3T",
                            "index": "0"
                        }
                    ],
        "signedLinePercentageString": "90",
        "stamp": {
            "bureauMarket": "Lloyd's",
            "bureauMarketCode": "4321",
            "bureauSubMarket": "",
            "businessUnit": " Lloyd's Syndicate IGC 4321",
            "stampType": "lloyds",
            "uniqueID": "ST579B30FD-1640-45AC…6FCD"
        },
```

```
        "uwRefs": [
          " ABC123456A20"
        ],
        "writtenLinePercentageString": "90"
      }
    ],
    "lineConditions": [],
    "sectionIdentifiers": [],
    "stampedAt": "2019-06-26 11:54:06",
    "subjectivities": [],
    "toStand": false
  }
],
```


control

The **control** object is an array of name-value pairs providing summary data about the contract, such as its status, the placing broker and the lead underwriter (see table Fig 3 below).

Variation: the **control** object does not form part of the Written Line (**RWWrittenLineSet**) or the Signed Line (**RWSignedLineSet**) documents.

Fig 3

Value	Description
contractType	Contract Type e.g. ‘Aviation Hull and Liability Insurance’
insuranceType	Insurance Type e.g. ‘Insurance’ or ‘Reinsurance’
insuredName	Name of the Insured
isFacility	Indicates if a risk record is a Facility or not
leadUnderwriter	Name of the Lead (Re)Insurer
leadUnderwriterChannel	Channel details of the Lead Underwriter
limitSummary	Currency and amount of the Policy Limit
originalDocID	ID of the document that the current risk record was created from

Value	Description
placementType	Indicates if a risk record is a Template, a Full Contract or a Off-Platform contract
placingBroker	Name of the Placing Broker
placingBrokerChannel	Channel details of the Placing Broker
status	Status of the contract, for example: ‘Draft’, ‘Firm Order’, ‘Signed’
umr	The UMR of the contract

Example JSON:


```
{
  "contractType": "",
  "insuranceType": "Insurance",
  "insuredName": "Linden Investments Limited",
  "isFacility": false,
  "leadUnderwriter": "Antares",
 "leadUnderwriterChannel": "antares_ALL",
  "limitSummary": "GBP 1,000,000",
  "originalDocID": "ICCE0647E4-0692...1F88::FO ",
  "placementType": "FullContract”,
  "placingBroker": "Price Forbes",
  "placingBrokerChannel": "priceforbes_ALL",
  "status": "Signed",
  "umr": " B0999ABC123456789"
}
```

What about PDF documents and XML formats?

A **JMRC** document can be converted to either PDF or Word document formats by applying transformation routines, and can also be mapped to the ACORD Global Placement Message (**GPM**) XML format, so that existing data and document standards can still be used.

What are the differences between JMRC and ACORD Global Placement Message (GPM)?

- **GPM** is fixed schema; **JMRC** is a semi-structured schema following the MRC document standard
- **GPM** dictates both the data formats required and defines sequence/order; **JMRC** does not impose any formatting or sequencing beyond that defined by the MRC
- **GPM** is XML, **JMRC** is JSON

A **GPM** has a clearly defined data structure where dates and numbers, for example, must conform to the standard. A **JMRC** holds the data in a looser way, but includes the entirety of the actual contract data and supports multiple date and number formats.

JMRC / GPM interoperability

A **JMRC** can be mapped to the **GPM** standard by coded rules, including those in the SDC guidelines, with no re-keying and no OCR requirements.

A **JMRC** (or indeed a complete contract document) cannot be reverse engineered from a **GPM** because the **GPM** dictates its own sequence and format on the data. A **GPM** does not hold the full contract wording or any contract specific variations or conditions that may be added to the contract by the Broker.

For example, if the contract advised a 'Limit GBP 5,000,000. From 31st December limit becomes GBP 6,000,000' the **GPM** would show the Limit as '5,000,000' and the Limit Currency as 'GBP'. The rest of this information, despite being present on the contract and of significance to the application of the contract, would not be present in a **JMRC** if it were to be reverse engineered *from a **GPM***, however it would be present in full, word for word, in a **JMRC** *created directly from an original contract*.

As a result Whitespace would recommend storing data as a **JMRC** document and generating a **GPM** from it, rather than vice-versa.

Reference Material

The MRC was created and is regularly improved by the LMG (see https://www.londonmarketgroup.co.uk/mrc) and provides a business document standard.

The JMRC is a simple JSON (see https://www.json.org) representation of the MRC to provide a fully digital contract record.

The Data Extraction and Mapping spreadsheet that is available as part of the LMTOM SDC Toolkit (https://tomsupports.london/structured-data-capture-toolkit - Technical Guides section) makes the possibility of text-and-extraction as powerful as an enforced-data-schema.