

# NEWS SERVICE REFINITIV DATA PLATFORM

## USER GUIDE

**DOCUMENT VERSION 2.0**

Date of Issue: June 2020

© [Refinitiv](#) 2020. All Rights Reserved. Republication or redistribution of Refinitiv content, including by framing or similar means, is prohibited without the prior written consent of Refinitiv. Refinitiv and the Refinitiv logo are trademarks of Refinitiv and its affiliated companies.

# Contents

|                                                        |           |
|--------------------------------------------------------|-----------|
| <b>About This Document .....</b>                       | <b>8</b>  |
| Feedback.....                                          | 8         |
| Support.....                                           | 8         |
| Your Personal Information.....                         | 9         |
| <b>1   Getting Started .....</b>                       | <b>10</b> |
| About the Refinitiv Data Platform .....                | 10        |
| Content Overview .....                                 | 10        |
| Learning Resources .....                               | 10        |
| Developer Community .....                              | 10        |
| API Playground.....                                    | 11        |
| <b>2   News Data Model.....</b>                        | <b>12</b> |
| Basic Concepts.....                                    | 12        |
| JSON Document Structure and Objects.....               | 12        |
| High-Level Representation of an RSF JSON Document..... | 12        |
| Message Header .....                                   | 15        |
| Item Set .....                                         | 15        |
| rightsInfo.....                                        | 16        |
| itemMeta.....                                          | 17        |
| contentMeta.....                                       | 22        |
| assert.....                                            | 25        |
| derivedFrom .....                                      | 26        |
| contentSet .....                                       | 27        |
| Provenance .....                                       | 28        |
| Additional Information.....                            | 29        |
| News Coding .....                                      | 29        |
| What is QCode .....                                    | 29        |
| QCode Construction.....                                | 29        |
| Product Codes.....                                     | 30        |
| Topic Codes .....                                      | 30        |
| Named Item Codes .....                                 | 31        |
| Attribution Codes.....                                 | 31        |
| RICs.....                                              | 31        |

|          |                                             |           |
|----------|---------------------------------------------|-----------|
|          | Company PermIDs .....                       | 32        |
|          | Extension Properties .....                  | 32        |
|          | Deduping .....                              | 33        |
|          | Sentiment .....                             | 33        |
|          | Confidence .....                            | 33        |
|          | Relevance.....                              | 34        |
|          | Social Tags.....                            | 35        |
|          | General Envelope Fields.....                | 35        |
| <b>3</b> | <b>About News Service.....</b>              | <b>37</b> |
|          | Overview.....                               | 37        |
|          | Content Format.....                         | 37        |
|          | Connectivity .....                          | 37        |
|          | Authentication.....                         | 37        |
|          | Token Types.....                            | 37        |
|          | Underlying Mechanics .....                  | 38        |
|          | Password Grant.....                         | 38        |
|          | Using the token .....                       | 39        |
|          | Refreshing the token .....                  | 40        |
|          | Token Scopes.....                           | 41        |
| <b>4</b> | <b>Request/Response Interface.....</b>      | <b>42</b> |
|          | Base URI .....                              | 42        |
|          | Paths and Operations.....                   | 42        |
|          | Responses.....                              | 42        |
|          | Analyze operation.....                      | 43        |
|          | Resolving ambiguity, usage of prefixes..... | 44        |
|          | Company prefixes .....                      | 44        |
|          | Instruments .....                           | 44        |
|          | RCS News Code .....                         | 45        |
|          | Source and Products.....                    | 45        |
|          | Free text search.....                       | 46        |
|          | Substitution symbols .....                  | 46        |
|          | SearchIn token .....                        | 46        |
|          | Data Range .....                            | 47        |

|                                             |    |
|---------------------------------------------|----|
| Boolean Range .....                         | 47 |
| OR usage .....                              | 47 |
| AND usage .....                             | 47 |
| NOT usage .....                             | 48 |
| Complex Boolean usage and parenthesis ..... | 48 |
| Common and useful codes .....               | 49 |
| Headlines operation.....                    | 49 |
| Request options .....                       | 49 |
| Query.....                                  | 49 |
| Limit .....                                 | 49 |
| Cursor .....                                | 49 |
| Sort Order.....                             | 49 |
| Relevancy.....                              | 49 |
| Response in details .....                   | 50 |
| Story ID.....                               | 50 |
| Version .....                               | 50 |
| Creator.....                                | 50 |
| Info Source .....                           | 50 |
| Language.....                               | 51 |
| Subject.....                                | 51 |
| Urgency .....                               | 52 |
| News creation date.....                     | 52 |
| News update date .....                      | 52 |
| Headline title.....                         | 52 |
| Headlines pagination.....                   | 52 |
| Images operation.....                       | 53 |
| Basic usage .....                           | 53 |
| Supported output formats .....              | 53 |
| Thumbnail generation .....                  | 53 |
| Metadata operations.....                    | 53 |
| Available operations .....                  | 54 |
| Metadata fields .....                       | 54 |
| Group Filter.....                           | 54 |
| accept-language header .....                | 55 |
| Metadata pagination .....                   | 55 |

|                                                    |           |
|----------------------------------------------------|-----------|
| Online Reports.....                                | 56        |
| Hierarchy of online reports.....                   | 56        |
| Curated report.....                                | 56        |
| Full Document Top 10 .....                         | 57        |
| Images.....                                        | 57        |
| Stories operation .....                            | 57        |
| HTML and JSON views.....                           | 57        |
| HTML view response.....                            | 57        |
| NewsWire response in details.....                  | 61        |
| Story content .....                                | 61        |
| Creator.....                                       | 61        |
| Urgency .....                                      | 61        |
| Info Source .....                                  | 61        |
| Language.....                                      | 62        |
| Subject.....                                       | 62        |
| News update date .....                             | 63        |
| News creation date.....                            | 63        |
| Title .....                                        | 63        |
| Clustering Keys .....                              | 63        |
| Suggest operation .....                            | 64        |
| Suggest Parameters .....                           | 64        |
| Query.....                                         | 64        |
| Items per group .....                              | 64        |
| accept-language header .....                       | 64        |
| Top News operations.....                           | 65        |
| Packages.....                                      | 65        |
| Headlines.....                                     | 65        |
| <b>5 News Messages on AWS.....</b>                 | <b>67</b> |
| Workflow to subscribe to News Messages on AWS..... | 68        |
| Login.....                                         | 68        |
| Alerts Subscription .....                          | 68        |
| Create the Credentials to a Cloud Queue .....      | 69        |
| Polling and Retrieving Messages.....               | 69        |
| Decrypting Messages .....                          | 69        |

|                                                                 |           |
|-----------------------------------------------------------------|-----------|
| Message Payload .....                                           | 70        |
| Output.....                                                     | 71        |
| Base URI .....                                                  | 72        |
| Paths and Operations.....                                       | 72        |
| Responses.....                                                  | 73        |
| <b>Appendix A - Glossary of Terms &amp; Abbreviations .....</b> | <b>74</b> |
| <b>Appendix B - Reuters Editorial Story Evolution .....</b>     | <b>75</b> |
| Simple Story .....                                              | 75        |
| Additional Takes .....                                          | 75        |
| Adding text.....                                                | 76        |
| Adding codes.....                                               | 76        |
| Adding text and codes.....                                      | 76        |
| Adding a cross-reference table .....                            | 76        |
| Normal Updates.....                                             | 77        |
| Replacing previous.....                                         | 77        |
| Appending latest.....                                           | 77        |
| Updates to Errors.....                                          | 77        |
| Refiles.....                                                    | 77        |
| Replacement .....                                               | 78        |
| Overwrite .....                                                 | 78        |
| Corrected and Correction.....                                   | 78        |
| Correcting Alerts.....                                          | 78        |
| Variations.....                                                 | 78        |
| Correcting Newsbreaks .....                                     | 79        |
| Corrected.....                                                  | 79        |
| Correction .....                                                | 79        |
| Variations.....                                                 | 79        |
| Withdrawals .....                                               | 80        |

## About This Document

News Services is a component of the Refinitiv Data Platform (RDP). RDP is a cloud-based service that provides access to Refinitiv content and services. RDP distribution layer enables standard REST-based APIs for you to explore. This guide provides details on the News API working environment and supporting resources. It is intended for software engineers and users of RDP who are familiar with the general principles of APIs.

This document is organized as follows:

- [Getting Started](#)
- [Story Construction and News Coding](#)
- [News Data Model](#)
- [About the RDP News API](#)
- [Appendixes](#)

## Feedback

We invite your comments, corrections, and suggestions about this document: access the [Feedback](#) option under [Help & Support](#) at [MyRefinitiv](#). Your feedback helps us continue to improve our user assistance.

## Support

The Refinitiv Statement of Service is available on [MyRefinitiv](#). MyRefinitiv is the Refinitiv portal that provides a single access point for timesaving support services, along with billing, user management, and information. For support, please raise a query by accessing [Help & Support](#) at [MyRefinitiv](#).

You are encouraged to [subscribe](#) to the following support channels to keep informed of changes to products and data, and to be notified of any service issues or changes:

### Change Notifications

- **Product** change notifications detail new, enhanced, or changed functionality, which may require your action, in products that you use.
- **Content** change notifications alert you to upcoming changes to real-time and historical data across all asset classes that are relevant to you.
- **RIC** change notifications inform you of planned changes to Reuters Instrument Codes.

### Service Alerts

You can subscribe to alerts about planned maintenance and unplanned service issues affecting your products and services and be notified via SMS or email.

## Your Personal Information

Refinitiv is committed to the responsible handling and protection of personal information. We invite you to review our [Privacy Statement](#), which describes how we collect, use, disclose, transfer, and store personal information when needed to provide our services and for our operational and business purposes.

# 1. Getting Started

## About Refinitiv Data Platform

Refinitiv Data Platform (RDP) is a cloud-based service, to provide unified access to Refinitiv content and services. RDP provides a single point of API-driven content and is backed by standard authentication and entitlements. RDP distribution layer enables a standard REST-based APIs for clients. The requested data is delivered using a mechanism, which is best suited for that particular data set.

## Content Overview

RDP provides a single point of REST API-driven content and discovery access to content and services backed by standard authentication and entitlements. Built on top of the cloud-based Refinitiv Data Platform, the APIs provide a standard workflow that enables clients – both humans and machines alike - to easily establish a session to access the platform and any content and services they are entitled to use.

One of the important News sources on RDP is Reuters News. Reuters has 2600 journalists in nearly 200 locations delivering international and national news with speed and insight. It delivers more than 2.2 million News Stories per year in 16 languages, actively covering 41,000 companies. Reuters has a long history of market-moving beats and exclusives related to macroeconomics, M&A activity, CEO changes, or corporate wrongdoing.

In addition to Reuters News, RDP also offers access to more than 40 third-party news sources that may be added to your Professional level subscription; there are more than 20 sources available for non-professional subscriptions. These sources provide corporate releases and market news covering several countries and languages. (A list of available sources available on the [Thomson Reuters Customer Zone](#); you will need a Customer Zone ID to access the list.).

RDP delivers Reuters and third-party News formatted for automated consumption via a modern, cloud-based platform and provides single interface access to the following facets of real-time News:

- Content - Refinitiv and third-party news are accessible from the same interface.
- Delivery - Delivery mechanism includes news streaming and request-response.
- Metadata - Stories contain rich metadata for accurate tagging and discoverability.
- Secure Access - You can access the content you are entitled to receive.
- Output Format - Modern and standard JSON output for easy consumption and parsing.

## Learning Resources

### Developer Community

The [Developer Community](#) portal provides RDP API documentation, forums and support.

The Developer Community portal provides access to the API Playground. This is an interactive Sandbox environment that allows users to experiment with individual API calls and their responses. See the below section for further details.

The portal enables you to connect with other API experts and users, and exchange information through community Q & A forums. You can browse or track issues and solutions for your relevant products.

## API Playground

The [API Playground](#) provides access to the RDP News APIs via a Graphical User Interface (GUI).

It acts as an interactive reference guide and tool which allows you to test and explore with the APIs, send requests, view responses, and review Swagger documentation.

The API Playground also provides a Swagger tab with documentation and a 'try it out' interface that returns responses in JSON from the application.

The API Playground URI is:

<https://apidocs.edp.thomsonreuters.com/Apps/ApiDocs>

Access to this site is available via a valid username and password, which you can obtain from your account manager. This site is best viewed using Google Chrome.

---

**Note:** It is recommended clients download the [News Service on Refinitiv Data Platform - Getting Started Guide](#) from the Developer Community portal. This is on the Documentation News Service General Guidelines page.

This guide contains more information on the API Playground and Developer Community, as well as general information on using the Refinitiv Data Platform

---

## 2. News Data Model

NewsML-G2 is an industry-wide XML specification for News Stories that is owned and maintained by the International Press and Telecommunications Council (IPTC). Generally, the news distributed from Thomson Reuters follows the RSF set of guidelines (Reuters Strategic Format), which is an implementation of NewsML-G2 XML format.

To meet the needs of RDP customers, Refinitiv has developed the RSF JSON format, which provides the rich metadata of RSF without the complexity of XML. RSF JSON is a literal representation of RSF; each of the name value pairs in JSON matches an RSF element or attribute.

### Basic Concepts

The main elements that describe the news item are as follows:

- [Headline](#) – the news item headline.
- [Content Set](#) – this contains the story payload (the full XHTML of the story).
- Two timestamps: (1) the [first created](#) date and time, and (2) the [version created](#) date and time. The first created timestamp is the date and time the first version of a Story was created, and it remains the same for all versions of that Story. The version created timestamp is the date and time when a particular version was created.
- [Content Metadata](#) - defines properties related to the content within, or referenced by, any of the primary items.
- [Provenance](#) - the details on how the metadata was added to the Story.

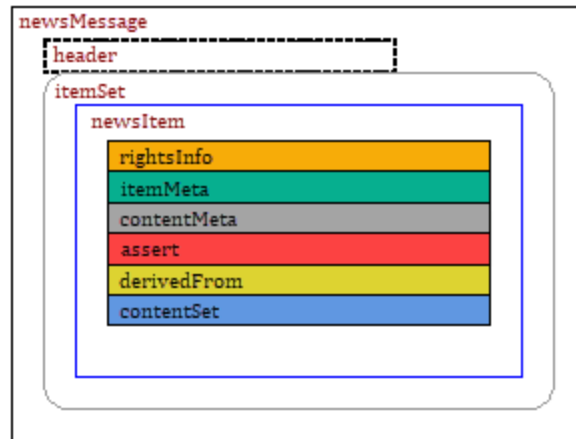
A Story also has a set of associated [category codes](#). First, you need to look at the [subject](#). The subject will have an identifier, which is a QCode; if any additional information relating to that concept has been provided in the file, it will be within the related [assert](#).

### JSON Document Structure and Objects

#### High-Level Representation of an RSF JSON Document

The document structure starts with the `newsMessage` container. It contains the Message Header (with transmission-related information), and the Item Set node that serves as a wrapper for a primary item, which, in this example, is a `newsItem` of type Text. The IPTC NewsML-G2 standard supports several other item types, but `newsItem` and `packageItem` are the only types currently available in RDP.

The `newsItem`, in turn, contains Rights Information, Item Metadata, Content Metadata, Assertions, Derivation Data and a Content Set (containing the Story Body).



The sections below describe each node and relevant fields in detail.

- [Message Header](#)
- [Item Set](#)
  - [rightsInfo](#)
  - [itemMeta](#)
    - [itemClass](#)
    - [provider](#)
    - [generator](#)
    - [versionCreated](#)
    - [firstCreated](#)
    - [pubStatus](#)
    - [role](#)
    - [title](#)
    - [signal](#)
    - [expires](#)
    - [link](#)
    - [itemMetaExtProperty](#)
  - [contentMeta](#)
    - [urgency](#)
    - [infoSource](#)
    - [creator](#)
    - [contributor](#)
    - [audience](#)
    - [altId](#)
    - [language](#)

- [keyword](#)
- [subject](#)
- [slugline](#)
- [headline](#)
- [assert](#)
- [derivedFrom](#)
- [contentSet](#)
  - [inlineXML](#)
  - [inlineData](#)

## Message Header

The message header information relates to the transmission of the enclosed Items, but may itself be extracted and stored for subsequent use.

The following table describes the fields under the Message Header:

| Object                   | Type            | Sample                                                                                       | Description                                                                                                                                                                                                                                      |
|--------------------------|-----------------|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sent</code>        | String          | <pre>"sent": {   "\$": "20170608T15:12:00.273Z" }</pre>                                      | The transmission timestamp of the item; typically set to the timestamp.                                                                                                                                                                          |
| <code>sender</code>      | String          | <pre>"sender": {   "\$": "reuters.com" }</pre>                                               | The party sending the items in the message, but not necessarily the originator of enclosed items.                                                                                                                                                |
| <code>transmitId</code>  | Number          | <pre>"transmitId": {   "\$": "tag:reuters.com,2019-06-08:newsml_Bw5CFMr4a:1" }</pre>         | A system-generated News Message GUID within the Provider's domain; this example contains the Story ID and version number in form ID:version.                                                                                                     |
| <code>priority</code>    | Number          | <pre>"priority": {   "\$": 3 }</pre>                                                         | The priority of the message; this differs from an item's editorial urgency. Priority varies from 1 to 9.<br>1 => High, 9 => Low. <ul style="list-style-type: none"> <li>• 1 - Alert</li> <li>• 2 - Newsbreak</li> <li>• 2&gt; - Story</li> </ul> |
| <code>destination</code> | String<br>Array | <pre>"destination": [   {     "_qcode": "NP:BSW"   },   {     "_qcode": "NP:CNR"   } ]</pre> | The destination where the message is sent. One or more product codes expressed as a QCode. For details, see <a href="#">Product Codes</a> .                                                                                                      |

## Item Set

In RDP, Item Set is one of the following:

- News Item (`newsItem`): A simple item of type Text.
- Package Item (`packageItem`): A set of references to other News Items and/or resources.

The following table describes the data fields under a `newsItem` or `packageItem`:

| Object                        | Type         | Sample                                                                                                                                                             | Description                                                                                                                                                                                                             |
|-------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>_conformance</code>     | String       | <code>"_conformance": "power",</code>                                                                                                                              | Conformance level; always <a href="#">power</a> for Reuters generated documents.                                                                                                                                        |
| <code>_guid</code>            | String       | <code>"_guid":<br/>"tag:reuters.com,2017-06-08:newsml_Bw5CFMr4a",</code>                                                                                           | System-generated Item GUID without the Item Version.                                                                                                                                                                    |
| <code>_standard</code>        | String       | <code>"_standard": "NewsML-G2",</code>                                                                                                                             | The individual standard within the IPTC G2-Standards (G2S).                                                                                                                                                             |
| <code>_standardVersion</code> | String       | <code>"_standardversion":<br/>"2.18",</code>                                                                                                                       | The G2S standard version.                                                                                                                                                                                               |
| <code>_version</code>         | Number       | <code>"_version": 6,</code>                                                                                                                                        | The version of the item. The version is updated by the publisher.                                                                                                                                                       |
| <code>catalogRef</code>       | String Array | <code>"catalogRef": [<br/>  {<br/>    "_href": "http://xml.media.reuters.com/g2-standards/catalogs/ReutersMedia_G2-Standards-Catalog_v1.xml"<br/>  }<br/>],</code> | Catalog Information that defines the Scheme References used by Scheme Aliases within a G2 document. A Remote Catalog Reference ( <a href="#">catalogRef</a> ) is included; the Reuters Catalog Reference is shown here. |

### rightsInfo

Contains copyright information and any usage terms set by a provider.

| Object          | Type            | Sample                                                                                                                                                                                                                                       | Description                                                         |
|-----------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| copyrightHolder | String<br>Array | <pre>"copyrightHolder": {   "literal": "Thomson Reuters" }</pre>                                                                                                                                                                             | The copyright holder of the Story, usually the source of the Story. |
| copyrightNotice | String<br>Array | <pre>"copyrightNotice": [   {     "\$": "Copyright Thomson Reuters"   } ]</pre>                                                                                                                                                              | Preferred copyright notice to use when publishing the Story.        |
| usageTerms      | String          | <pre>"usageTerms": [   {     "\$": "This file is provided for EAP sample purposes only; it's structure and detail is subject to change, and should not be used as a definitive reference for actual development and processing."   } ]</pre> | Usage terms or caveats, if any, set by the source.                  |

### itemMeta

The Item Meta defines properties related to any of the Primary Items in its entirety, as opposed to the content within, or referenced by that item.

| Object                      | Type         | Sample                                                                                                           | Description                                                                                                                                                                                                                                                                                                                   |
|-----------------------------|--------------|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>itemClass</code>      | String       | <pre>"itemClass": {   "_qcode": "ninat:tex",   "_rtr:msgType": "A" }</pre>                                       | Indicates the content included/referenced by the item.                                                                                                                                                                                                                                                                        |
| <code>provider</code>       | String       | <pre>"provider":{   "_literal":     "reuters.com" }</pre>                                                        | The party responsible for management and release of the item (but not necessarily the content conveyed by the item).                                                                                                                                                                                                          |
| <code>generator</code>      | String Array | <pre>"generator": [ {   "\$":"UCDP:parsn_bwnx_10.5 1.13.5_1.2.31404:",   "_versioninfo":     "1.0.31404" }</pre> | Information about the system that generated the item. Note that “generator” names may change and should not be relied on to uniquely identify content or systems.                                                                                                                                                             |
| <code>versionCreated</code> | String       | <pre>"versionCreated": {   "\$": "2019-06-08T15:12:00.273Z" }</pre>                                              | The date and time that this item was last updated. The timestamp is always UTC as indicated by time zone indicator ‘Z’.                                                                                                                                                                                                       |
| <code>firstCreated</code>   | String       | <pre>"firstCreated": {   "\$": "2017-06-08T15:12:00Z" }</pre>                                                    | <p>The date and time that this item was initially created. It remains the same for all versions of this item.</p> <p>The combination of PNAC and <code>firstCreated</code> is used in conjunction with a separate version as the Story is published. The timestamp is always UTC as indicated by time zone indicator ‘Z’.</p> |

| Object                 | Type   | Sample                                                                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------|--------|---------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>pubStatus</code> | String | <pre>"pubStatus": {   "_qcode":   "stat:usable" }</pre>                                           | <p>Defines the publication status for the particular version of the item.</p> <p>Indicates whether an item is currently usable. Valid QCodes are:</p> <ul style="list-style-type: none"> <li>• <code>stat:canceled</code></li> <li>• <code>stat:usable</code></li> <li>• <code>stat:withdrawn</code></li> </ul> <p>The <code>pubStatus</code> is normally <code>"stat:usable"</code> for most content.</p> <ul style="list-style-type: none"> <li>• When <code>pubStatus</code> is <code>"stat:canceled"</code>, consumers must remove all versions of the item from website, caches, etc. The item may be retained for archival purposes only.</li> </ul> |
| <code>role</code>      | String | <pre>"role": {   "_qcode":   "itemRole:N" }</pre>                                                 | <p>The role that the item is performing. If omitted, it is assumed to be a News Story. Valid values are:</p> <ul style="list-style-type: none"> <li>• A – Alert</li> <li>• B – Newsbreak</li> <li>• N – News Story/Normal</li> <li>• U – Bulleting</li> <li>• T – Additional Take</li> <li>• F – Feature</li> </ul>                                                                                                                                                                                                                                                                                                                                        |
| <code>title</code>     | String | <pre>"title": [   {     "\$": "2017 Report on Life Sciences Market Research Industry"   } ]</pre> | <p>The document title, which is the a one-line description of the item. The title field is just the text of the Headline without another headline information (such as Headline Tag markup).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Object  | Type            | Sample                                                                                                                                                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| signal  | String<br>Array | <pre>"signal": [   {     "_qcode":     "sic:CPX"   },   {     "_qcode":     "prodId:TXT"   },   {     "_qcode":     "pmt:text"   },   {     "_qcode":     "edStat:N"   } ]</pre> | <p>A signal is a high-level indication to a G2 processor, implying that some specific processing/filtering may be required. Signals can be broadly split into the following categories:</p> <ul style="list-style-type: none"> <li>• Explicit – Used for filtering/routing</li> <li>• Compound – A shorthand way of conveying some more complex metadata combination.</li> </ul> <p>A signal is easier for a processor to handle than a complex metadata combination because a single flag can be checked. Also, the signal decouples the consumer from any changes in the business logic associated with the signal.</p> |
| expires | String          | <pre>"expires": [   {     "\$": "2018-07- 08T15:12:00.273Z"   } ]</pre>                                                                                                          | The date the item expires.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

| Object | Type            | Sample                                                                                                                                                                                                                                                                                               | Description                                                                                                                                                                                                  |
|--------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| link   | String<br>Array | [<br>{<br>"_how":<br>"howextr:tool",<br>"_rel":<br>"irel:wasPackaged<br>In",<br>"_residref":<br>"urn:newsml:reute<br>rs.com:20020924:S<br>PDOC_55977242002"<br>,<br>"_why":<br>"why:inferred",<br>"instanceOf": [<br>{<br>"qcode":<br>"RR:1305",<br><br>"_type":<br>"cptType:12"<br>}<br>]<br>}<br>] | Links to other resources, indicating the relation of the other resource to this item; this example shows that this Item was <i>packaged</i> in the Item identified by the <u><a href="#">_residref</a></u> . |

| Object                           | Type   | Sample                                                                                                                                                                                                                                                                                                                                                                                                  | Description                                                                                                                                                                                                                                                                  |
|----------------------------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>itemMetaExtProperty</code> | String | <pre> "itemMetaExtProperty": [   {     "_creator": "rft ResRef:sys24",     "_id": "mdx1",     "_rel": "extCptR el:hasMduMinorVersio n",     "_value": "2",     "_valuedatatype": "xs:positiveIntege r",     "related": [       {         "_rel": "ext CptRel:hasRelatedTim estamp",         "_value": "2 020-01- 14T18:38:06.453",         "_valuedatat ype": "xs:dateTime"       }     ]   } ], </pre> | <p>Defines an extension property associated with the Item Metadata. For details, see <a href="#">Extension Properties</a>.</p> <p>This example shows that this Item has an Mdu Minor Version value of 2, which itself has a related Timestamp of 2020-01-14T18:38:06.453</p> |
| <code>versionedId</code>         | String | <pre> "rtr:versionedId": [   {     "_guid": "tag:reuters.com,202 0-01- 14:newsml_RTVP14q: 9"   } ] </pre>                                                                                                                                                                                                                                                                                               | <p>Provides a GUID which is the concatenation of the Item GUID and the Item Version.</p>                                                                                                                                                                                     |

### contentMeta

The Content Meta group defines properties related to the content within, or referenced by, any of the primary items. This group contains administrative properties ([urgency](#) etc.) and descriptive properties ([language](#), [subject](#) etc.).

| Object                   | Type            | Sample                                                                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>urgency</code>     | Number          | <pre>"urgency": {   "\$": 3 }</pre>                                                                                                    | <p>The Editorial urgency of an individual Primary Item; this differs from the transmission priority (header/priority).</p> <p>Valid values: 1-9.</p> <p>Default value: 4.</p> <p>Note that the header priority of the message should reflect the highest urgency (lowest numeric value) of any primary items carried by the news message.</p>                                                                                                                                                                    |
| <code>infoSource</code>  | String<br>Array | <pre>"infoSource": [   {     "_qcode": "NS:RTRS"   },   {     "_qcode": "NS:RTRS"   },   {     "_role": "cRole:origProv"   } ]</pre>   | <p>An Info Source is a Party which <i>originated, distributed, aggregated or supplied</i> the content or <i>provided</i> information used to <i>create or enhance</i> the content.</p> <p>Common values:</p> <p><code>cRole:source</code>: the original content owner.</p> <p><code>cRole:origProv</code>: the original object provider.</p> <p>Values can be NS:RTRS for Thomson Reuters content or an alternative third-party attribution code.</p> <p>For details, see <a href="#">Attribution Codes</a>.</p> |
| <code>creator</code>     | String<br>Array | <pre>"creator": [   {     "_qcode": "NS:RTRS"   },   {     "_role": "cRole:source"   } ]</pre>                                         | <p>The organization that created the Story.</p> <p>If the content creator is also the source, then this should be stated. (<code>@role="cRole:source"</code>).</p>                                                                                                                                                                                                                                                                                                                                               |
| <code>contributor</code> | String<br>Array | <pre>"contributor": [   {     "_qcode": "NS:RTRS"   },   {     "_role": "cRole:enhancer"   },   {     "_role": "cRole:rep"   } ]</pre> | <p>The party that modified or enhanced the content. It usually includes levels of details.</p> <p>Common values:</p> <p><code>cRole:enhancer</code> - The party that enhanced the content with additional metadata.</p> <p><code>cRole:rep</code>: The party that reported the content.</p>                                                                                                                                                                                                                      |

| Object                   | Type            | Sample                                                                                                                                                                                                          | Description                                                                                                                                                                      |
|--------------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">audience</a> | String<br>Array | <pre>"audience": [   {     "_qcode": "NP:RITV"   } ]</pre>                                                                                                                                                      | <p>The audience(s) for which the item is of particular interest.</p> <p>It is identified by Product Codes.</p> <p>For more information, see <a href="#">Product Codes</a>.</p>   |
| <a href="#">altId</a>    | String          | <pre>"altId": [   {     "_type": "idType:USN",     "\$": "RTV1CP14q"   } ]</pre>                                                                                                                                | <p>Enables alternative identifiers to be defined.</p> <p>Common Values:</p> <ul style="list-style-type: none"> <li>- USN – USN.</li> <li>- EAID - Editorial Asset Id.</li> </ul> |
| <a href="#">language</a> | String<br>Array | <pre>"language": [   {     "_tag": "en"   } ]</pre>                                                                                                                                                             | <p>The language(s) relating to content within the item (not the subject language).</p> <p>ISO-639 language code.</p>                                                             |
| <a href="#">keyword</a>  | String          | <pre>"keyword": [   {     "_creator": "rftRes Ref:sys24",     "_how": "howextr:to ol",     "_id": "s3m160",     "_role": "descRole :socialTag",     "_why": "why:infer red",     "\$": "Chase Bank"   } ]</pre> | <p>Keywords often contain terms which relate to the content in ways other than aboutness.</p>                                                                                    |
| <a href="#">subject</a>  | String          | <pre>"subject": [   {     "_qcode": "B:161"   },   {     "_qcode": "M:32"   },   {     "_qcode": "G:4"   }, ]</pre>                                                                                             | <p>Topic codes and entity identifiers (such as organizations) that relate to this new item.</p> <p>For details, see <a href="#">Topic Codes</a> and <a href="#">RICs</a>.</p>    |

| Object                              | Type            | Sample                                                                                                                                                                                                                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>slugline</code>               | String<br>Array | <pre>"slugline": [   {     "_confidence": 50,     "_creator": "rftRes Ref:sys24",     "_how": "howextr:to ol",     "_id": "s9m960",     "_rank": 1,     "_role": "newslineR ole:auto",     "_separator": "-",     "_why": "why:infe rr ed",     "\$": "JPMORGAN- RESULTS/"   } ]</pre> | <p>Sluglines are a string of keywords applied to News Stories, pictures and videos. The slugline represents the main event that the News Story is related to.</p> <p>Each Reuters News Story has only one slugline.</p> <p>A Slugline may have a <code>_role</code> attribute if it is not a Reuters Story; if multiple Sluglines are provided, then the priority is defined by the <code>_rank</code> attribute.</p>                                                                                                                                                                     |
| <code>headline</code>               | String<br>Array | <pre>"headline": [   {     "_dir": "ltr",     "_xml:lang": "en",     "\$": "Reuters Insi der - Trading rebound b oosts JP Morgan, Citi"   } ]</pre>                                                                                                                                    | <p>The headline text of the News Item along with a language (<code>_xml:lang</code>) and text direction indicator (<code>_dir</code>).</p> <p>Headlines that are naturally longer than 512 bytes are truncated by Thomson Reuters. If truncated, then multi-byte characters are preserved. For example, if the last character uses bytes #512 and #513, then the Headline will be truncated after the 511th byte.</p> <p>Some third-party news sources truncate their headlines at fewer than 512 bytes. Truncated headlines have “...” appended, to denote that they were truncated.</p> |
| <code>contentMetaExtProperty</code> | String          |                                                                                                                                                                                                                                                                                        | <p>Defines an extension property associated with the Content Metadata. For details, see <a href="#">Extension Properties</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

## assert

An assertion is used to include partial information relating to a concept, such as a person, place, organization, topic, etc. The assertion enables the partial information to be defined once within a document and referenced by multiple metadata instances in the same document via the QCode associated with the Assert.

| Object              | Type         | Sample                                                                                                                                                                                                                                                                                                                                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>assert</code> | String Array | <pre>//Content Coding "subject" : [   {     "_qcode": "B:195"   } ] . . . //Corresponding Assert "assert":[   {     "_qcode" : "B:195"     "sameAs": [       {         "_qcode": "N2:BSUP"         "name": [           {             "\$":"Business Support Services (TRBC level 4)"           }         ]       }     ]     "type": [       {         "_qcode": "cptType:2"       }     ]   } ]</pre> | <p>Provides details and descriptions for associated QCodes.</p> <p>The example in the previous column is an assertion for RCS QCode "B:195". It shows the News 2000 code, code name, and <code>cptType</code>. <code>cptType</code> attribute provides the type of metadata.</p> <p>Common Values:<br/> <code>cptType:8</code> – Organization<br/> <code>cptType:9</code> – Person.</p> |

### derivedFrom

Metadata values are not always added explicitly by human interaction but could be added by an automated derivation. In this case, it could be valuable to indicate the concept(s) from which a specific value of a metadata property has been derived. For this purpose, the optional `derivedFrom` elements can be used.

The QCode of the `derivedFrom` element defines the concept from which another concept has been derived. The `idrefs` attribute of this element refers to the `id` attributes of all properties in this item whose value has been derived from the concept represented by `derivedFrom`.

In the example below, the concept with `id` = 'A1' was derived from the concept 'NP:RITV'

```
"derivedFrom": [
  {
    "_idrefs": "A1",
```

```
    "_qcode": "NP:RITV"
  },
  {
    "_idrefs": "A1",
    "broader": [
      {
        "_qcode": "mduAction:enrichMetadata"
      }
    ],
    "related": [
      {
        "_rel": "extCptRel:hasworker",
        "related": [
          {
            "_qcode": "rftResRef:sys26",
            "_rel": "skos:broader"
          }
        ]
      }
    ]
  }
],
},
...
```

#### contentSet

Contains the actual Story (payload), either structured content or plain text.

| Object                  | Type | Sample                                                                                                                                                                                                                                                                                                                                                            | Description                                                                                                                                 |
|-------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <code>inlineXML</code>  |      | <pre>"inlineXML": [   {     "_contenttype":     "application/xhtml+xml",     "\$": "[body of XML document with XHTML markup]"   } ]</pre>                                                                                                                                                                                                                         | A block of XML with a given format. For RSF, the format must be XHTML.                                                                      |
| <code>inlineData</code> |      | <pre>"inlineData": [   {     "_contenttype":     "text/plain",     "_dir": "ltr",     "_xml:lang": "en",     "\$": "For best results when printing this announcement, please click on link below:\r\nhttp://pdf.reuters.com/h tmlnews/htmlnews.asp?i=43059c3bf 0e37541&amp;u=urn:newsml:reuters.com:2 0170608:nBw5CFMr4a\r\n\r\n\r\n2017 [text body]"   } ]</pre> | A block of free text with no structured markup. This field may also include (encoded) binary data, such as PDFs, images, or Word documents. |

## Provenance

The details on how the metadata was added to the Story is called provenance (meta-metadata) and is provided in several fields:

- `"_creator"` – indicates what system added the metadata.
- `"_how"` – indicates whether it was a tool that added the metadata. If it was a tool, then the `"id"` attribute refers to [derivedFrom](#) element(s) that provide further levels of details.
- `"_why"` – reason for application.
- `"_id"` – unique identifier for this metadata used across this particular Story only to track provenance information. In the example below, `s3` is the identifier for `B:126` and `grp2` is the identifier for `hml1nd:high`.
- `"_rel"` – provides the relationship of this piece of data to the parent data. In the example below, the `"hasRelevanceGroup"` shows that of the metadata `"B:126"` has an associated Relevance Group of 'High'.

### Example:

```
{
  "_creator": "rftResRef:sys26",
  "_how": "howextr:tool",
```

```

    "_id": "s3",
    "_qcode": "B:126",
    "_why": "why:inferred",
    "related": [
      {
        "_creator": "rftResRef:sys24",
        "_how": "howextr:tool",
        "_id": "grp2",
        "_qcode": "hmlInd:high",
        "_rel": "extCptRel:hasRelevanceGroup",
        "_why": "why:inferred"
      }
    ]
  }
}

```

## Additional Information

### News Coding

All News Items, including Reuters News and third-party, are tagged with category codes that describe the content of the Story. The preferred Reuters coding scheme is RCS (Reuters Classification Scheme), and you will find RCS codes throughout the JSON document in properties such as audience, destination, and subject. All the available RCS codes are described in the tables throughout this section. RCS uses QCode construction.

#### What is QCode

A QCode - which stands for Qualified Code - is a short format of a Uniform Resource Identifier (URI).

URIs are used by Semantic Web technology to identify resources, which can be abstract concepts, persons, organizations, companies, locations, objects, events and much more.

A URI could be <http://cv.iptc.org/newscodes/mediatopic/20000309> and this may be too long for practical use.

A corresponding QCode medtop:20000309 is shorter and more convenient; it has the format *alias*":"*code*.

#### QCode Construction

The QCode string is composed of three parts – {*alias*} {*separator*} {*value*} – and all three parts must be present:

1. {*alias*}: *Scheme Alias* or prefix, for example, "inat".
2. {*separator*}: *Scheme Code Separator* is always a colon ":".
3. {*value*} *Code Value*, for example, "text".

The Scheme Alias resolves to a Scheme URI. The Code Value is appended to the Scheme URI to produce the corresponding Concept URI. The following examples illustrate step-by-step how QCodes are constructed:

#### Example 1: A Scheme from the IPTC - Indicating an Item is a Text Item

- **Scheme: News Item Nature, a scheme from the IPTC.**
- Scheme URI: <http://cv.iptc.org/newscodes/ninature/> (\*1)
- Scheme Alias: {alias} ::= 'ninat'
- Code Value {value} ::= 'text' (Mastered by the IPTC)
- Concept URI is <http://cv.iptc.org/newscodes/ninature/text> (\*1)
- Resulting QCode: {qcode} ::= 'ninat:text' (\*4)

#### Example 2: An 'External' Scheme from the IPTC - Indicating an SEC Submission Type

- **Scheme: SEC Submission Type.**
- Scheme URI: <http://cvx.iptc.org/secst/> (\*1)
- Scheme Alias: {alias} ::= 'secst'
- Code Value: {value} ::= '40-OIP/A'
- Concept URI: <http://cvx.iptc.org/secst/40-OIP/A> (\*2)
- Resulting QCode: {qcode} ::= 'secst:40-OIP/A' (\*4)

### Product Codes

Product Codes, also known as News Service Codes, identify which news products the News Item is associated with. They are typically tailored to specific audiences.

| Example                                          | Identification                                                                                                                                                         |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>NP:M</i> for Money International News Service | As <code>_qcode</code> in <code>audience</code> and <code>destination</code> fields.<br>Values are prefixed by <b>NP</b> : The Scheme Alias for <i>News Services</i> . |
| <i>NP:FB</i> for the French General News Service |                                                                                                                                                                        |
| <i>NP:CNR</i> for Corporate News Releases        |                                                                                                                                                                        |

### Topic Codes

Topic codes describe a Story's subject and give basic metadata information about the Story. Topic codes include asset classes, geographies, events, industries/sectors, and other types.

| Example                                | Identification                                                                                                                                                     |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A:G for Credit Default Swaps           | As <code>_qcode</code> in <code>subject</code> field. Values are prefixed by the Scheme Alias for a given topic type.                                              |
| B:50 for Environmental Services Sector | For example,<br><b>A:</b> for Asset Class                                                                                                                          |
| E:F for Dividends                      | <b>B:</b> for Business Sector<br><b>E:</b> for Events                                                                                                              |
|                                        | In rare instances, the topic type is changed after topic creation. In these cases, the topic keeps the alias of the original type.                                 |
|                                        | The equivalent News2000 QCodes for the RCS QCodes are given in the <code>assert</code> object; in this example they are N2:CDV, N2:ENVS, and N2:DIV, respectively. |

### Named Item Codes

Named Item Codes (NICs), or recurring reports, identify the News Items that are regularly published and follow a pattern. Items sharing the same code cover periodic Updates to the same subject matter and often have similar Headlines.

| Example                                            | Identification                                                                                                       |
|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| NI:L or RR:1200 for news on UK stocks              | As <code>_qcode</code> , with prefix <b>RR:</b> for the Recurring Report code or <b>NI:</b> for the Named Item Code. |
| NI:MEAL/DEL or RR:2932 for CBOT Soymeal Deliveries |                                                                                                                      |

### Attribution Codes

Attribution Codes, or News Source Codes, identify the source of the Story, either Refinitiv or a third-party provider.

| Example                  | Identification                                                                                                          |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------|
| NS:RTRS for Reuters News | As <code>_qcode</code> in <code>infoSource</code> . Values are prefixed by <b>NS:</b> the Scheme Alias for News Source. |
| NS:BSW for Business Wire |                                                                                                                         |

### RICs

Reuters Instrument Codes (RICs) represent companies, economic indicators, or foreign exchange pairs.

| Example                                                   | Identification                                                  |
|-----------------------------------------------------------|-----------------------------------------------------------------|
| <i>R:IBM.N</i> for IBM                                    | As <code>_qcode</code> , with R: prefix in <code>subject</code> |
| <i>R:EURGBP=</i> for the Euro/British Pound exchange rate |                                                                 |
| <i>R:USNAHB=ECI</i> for the NAHB Housing Market Index     |                                                                 |

### Company PermlDs

Thomson Reuters is supplementing its identification of companies with an organizational-level identifier called the “PermlD”. Since RICs are quote identifiers, they are less stable over time than a company identifier.

Furthermore, the many-to-one relationship between quotes and companies makes it possible also for multiple RICs of a single company to be tagged to a single News Item. Thomson Reuters is thus exposing its privately mastered information model and making the PermlD publicly available. See <https://permid.org> for more information on the PermlD.

Most company RICs in the live Real-time News data will be supplemented with a PermlD.

| Example                     | Identification                                                               |
|-----------------------------|------------------------------------------------------------------------------|
| <i>P:4295904307</i> for IBM | As <code>_qcode</code> in <code>assert</code> field, prefixed by <b>P:</b> . |

### Extension Properties

Most News elements that wrap a group of metadata properties have a generic extension property named after the parent element suffixed with “`ExtProperty`”, for example `<contentMeta>` has `<contentMetaExtProperty>`. This enables a special-use property to be expressed using a common type which includes `_rel` to define the relationship with the target concept.

Example:

This Item (the parent of `contentMeta`) has a Dedupe Key of value

“`dedupeKey:strict:20200114_34146d18269ff43d17383124345abab722cb8fe1`”, and that Key has a Key Type of “`dedupeKeyType:strict`”

```

"contentMetaExtProperty": [
  {
    "_creator": "rftResRef:sys24",
    "_how": "howextr:tool",
    "_id": "s1m5",
    "_qcode":
"dedupeKey:strict:20200114_34146d18269ff43d17383124345abab722cb8fe1",
    "_rel": "extCptRel:hasDedupeKey",
    "_why": "why:inferred",
    "related": [

```

```

    {
      "_creator": "rftResRef:sys24",
      "_how": "howextr:tool",
      "_qcode": "dedupeKeyType:strict",
      "_rel": "extCptRel:hasDedupeKeyType",
      "_why": "why:inferred"
    }
  ]

```

## Deduping

The deduplication keys are used to track the deduplication of content within a particular Story. The example below defines that the document has a dedupe key and the value of this dedupe key is the QCode.

Example:

```

{
  "_creator": "rftResRef:sys24",
  "_how": "howextr:tool",
  "_id": "s2m190",
  "_qcode": "dedupeKey:event:20200114_C266999c",
  "_rel": "extCptRel:hasDedupeKey",
  "_why": "why:inferred",

```

## Sentiment

The sentiment of a piece of text is its positivity or negativity. It gives an objective idea of whether the text uses mostly positive, negative, or neutral language.

Sentiment may be provided at two different levels:

- at the document level (the News Story).
- at the company level, if the company is mentioned in the Story. Provided under the `assert` field.

The sentiment has three different parts (`hasSentimentPositive`, `hasSentimentNegative`, `hasSentimentNeutral`). The positive, negative and neutral values represent the proportion of text that falls in these categories. All these values should sum up to 1. The value needs to be interpreted based on the `_valuedatatype` property.

Example:

```

  "_rel": "extCptRel:hasSentimentNegative",
  "_value": "0.689233",
  "_valuedatatype": "xs:decimal",

```

**Note:** Sentiment is available to customers subscribing to premium Analytics

## Most Read

XXXXXX.

Most Read Sentiment may be provided at two different levels:

Example:

```
"_rel": "extCptRel:hasSentimentNegative",
  "_value": "0.689233",
  "_valuedatatype": "xs:decimal",
```

Note: Most Read analytics is available to customers subscribing to premium Analytics

## Confidence

The `confidence` attribute expresses the provider's confidence in the information provided. A value of 100 indicates the highest confidence.

Example:

```
"slugline": [
  {
    "_confidence": 50,
    "_creator": "rftResRef:sys24",
    "_how": "howextr:tool",
    "_id": "s9m960",
    "_rank": 1,
    "_role": "newslineRole:auto",
    "_separator": "-",
    "_why": "why:inferred",
    "$": "JPMORGAN-RESULTS/"
  }
]
```

## Relevance

This indicates the relevance of the metadata to the news content to which it is attached. The relevance can be high, low or medium. A high relevance indicates that the metadata truly expresses the essence of the content, while a low relevance indicates a low correlation between the metadata and the content. We provide relevance for the following types of metadata:

- topic codes
- organizations
- subjects
- audience

Example:

```
"subject": [  
  {  
    "_creator": "rftResRef:sys26",  
    "_how": "howextr:tool",  
    "_id": "s1",  
    "_qcode": "A:1",  
    "_why": "why:inferred",  
    "_relevance": 100,
```

## Social Tags

Social tags are dynamic topics in the world of Wikipedia, and they describe what the News Story is about as a whole.

### Example:

```
"keyword": [  
  {  
    "_creator": "rftResRef:sys24",  
    "_how": "howextr:tool",  
    "_id": "s3m160",  
    "_role": "descRole:socialTag",  
    "_why": "why:inferred",  
    "$": "Chase Bank"  
  },
```

## General Envelope Fields

The news messages are packaged in a general envelope, which contains the news object holding news specific information.

The data model of each message is as follows:

| Field                        | Description                                                                          |
|------------------------------|--------------------------------------------------------------------------------------|
| Attributes                   | Attributes that describe the transaction: the name of content, the type of data.     |
| Version                      | Provided by the publisher, controlled by the EDP Alerts team.                        |
| Heartbeat                    | Only populated when this is a heartbeat message.                                     |
| ECP Message ID               | Provided by the publisher.                                                           |
| Source Sequence Number       | Provided by the publisher; optional for heartbeat message.                           |
| Distribution Sequence Number | Provided by the EDP content distributor per Alert subscription.                      |
| ECP Distribution ID          | This is the ECP distribution ID.                                                     |
| Source Timestamp             | Publisher timestamp provided by the publisher.                                       |
| Receive Timestamp            | EDP received timestamp.                                                              |
| Distribution Timestamp       | Date/time that EDP posts the message to the user's queue.                            |
| Payload Version              | The version of the payload content, version control by the source publisher.         |
| Subscription ID              | The Alert subscription ID.                                                           |
| Href                         | A link to retrieve the message content in case the payload is larger than the limit. |
| Payload                      | The actual message content; in the case of news, the news JSON object.               |

## 3. About News Services on RDP

### Overview

The data from an RDP service can be delivered in variety of mechanisms, depending on content set. Following are the content delivery types:

- **Request - Response** - is the most common type, where the data is delivered through a RESTful web service. An application uses a web request (HTTP GET, POST, PUT or DELETE) to convey the request message and parameters, and the RDP service responds with data in a synchronous manner.
- **Alerts delivery** is a mechanism to receive *asynchronous updates* (Alerts) to a subscription criterion.

### Content Format

The input for the RDP News interfaces is a standard predefined JSON format. The output is encoded UTF-8 JSON format. Details on JSON structure and format are provided in the [News Data Model](#) section of this document.

### Connectivity

The RDP News API is accessible over the public internet. If you have access to the internet from your site, you can access the RDP News API via the following standard outbound methods and ports:

- HTTP (port 80)
- HTTPS (port 443)

### Authentication

Refinitiv Data Platform entitlement check is based on OAuth 2.0 specification. The first step of an application workflow is to get a token, which will allow access to the protected resource.

### Token Types

OAuth system defines various mechanisms to get an access token. These mechanisms are called *Grant type*. An application can get an access token, initially through *Password Grant* and later by using a *Refresh Token*. A successful request for a token will always return two tokens:

1. A short-lived **Access Token**, which expires in a few (5) minutes. This token is used in every subsequent REST API call until it expires. Access token also have an implicit scope(s) and can only be used for products that it is scoped for. So, the scope of REST API which the user is trying to invoke, must be included when requesting an access token.
2. A long-lived **Refresh Token**. This token is only used to get new Access and Refresh tokens after the old one has expired. It cannot be used to access protected resource, i.e. cannot be used with data API's.

A successful authorization call generates new Access and Refresh tokens.

Initially when an application does not have any token, it must use a username/password combination to get the first set of tokens. This type of request is called **Resource Owner Password Credentials** or **Password Grant** in short. Once an application has a set of tokens, it should store the Refresh token in a persistent storage and use the Access token to get data. Upon expiry of Access token, the Refresh token must then be used to get new Access and Refresh tokens. This use of Refresh token to get subsequent tokens, avoids having to send password across the network repeatedly.

### Underlying Mechanics

We will briefly discuss the HTTP requests/response for authorization. For Refinitiv Data Platform, the token endpoint is <https://api.refinitiv.com/auth/oauth2/VERSION/token>, where the <VERSION> number will change, as the API evolves.

There are few additional parameters which are required by token API, for compliance with OAuth 2.0 specification:

| Parameter     | Description                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Grant Type    | Defines whether you requesting new tokens using username/password combination ( <b>password</b> ), or through an existing refresh token ( <b>refresh_token</b> ).                                                                                                                                                                                                                              |
| Username      | The username used for "Password Grant". This is typically an email address, or a Machine ID which has been assigned to you.                                                                                                                                                                                                                                                                    |
| Password      | Password associated with the username above. This field may contain special characters or reserved characters which may have to be escaped. The applications must be able to handle special characters.                                                                                                                                                                                        |
| Client ID     | Aka Application ID, aka AppKey is a unique identifier, defined for the application or the user making a request. The user can generate/manage their Application ID's using the AppkeyGenerator. The client ID parameter can be passed in the request body or as "Authorization" request header which is encoded as base64. This Authorization header method is used in the interactions below. |
| Client Secret | Part of OAuth specification, this parameter is <b>not</b> used in RDP.                                                                                                                                                                                                                                                                                                                         |
| Token Scope   | The user can optionally limit the scope of generated token, so that Access token is valid only for a specific data set.                                                                                                                                                                                                                                                                        |

A successful authentication response from the server contains the following parameters:

| Parameter     | Description                                                                                                   |
|---------------|---------------------------------------------------------------------------------------------------------------|
| access_token  | The token used to invoke REST data API calls as described above.                                              |
| refresh_token | Refresh token used for getting the next access token.                                                         |
| expires_in    | The Access token validity time in seconds.                                                                    |
| scope         | A list of all the scopes this token can be used with ( <b>trapi.data.news.read</b> is used for News content). |
| token_type    | "Bearer" - i.e., RDP token service has generated and sent us the Access token in the response message.        |

### Password Grant

This is a Request using *Password Grant*, where the user is logging in with their credentials. Note, that for a very first-time login, when a password needs to be changed, **newPassword** should be included in the request as well.

```
POST http://api.refinitiv.com/auth/oauth2/v1/token HTTP/1.1
```

```
Host: api.refinitiv.com
```

```
User-Agent: python-requests/2.12.4
```

```
Accept-Encoding: gzip, deflate
```

```
Accept: application/json
```

```
Connection: keep-alive
```

```
Content-Length: 112
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Authorization: Basic
```

```
RDZENz*****DNKNEREQ3NDo=username=****&password=****&grant_type=password&scope=trapi
```

Also note, that the client ID has been sent in the HTTP header as Basic Authentication **Authorization: Basic RDZENz\*\*\*\*\*DNKNEREQ3NDo=**. This could have been included in the request body as well. If our credentials are valid, the server will respond with a message:

```
HTTP/1.1 200 OK
```

```
Date: Thu, 05 Apr 2018 18:57:06 GMT
```

```
Content-Type: application/json
```

```
Content-Length: 1604
```

```
Connection: keep-alive
```

```
X-Tr-Requestid: dd6e2131-917d-4cd8-856e-3754ff6a6990
```

```
{
  "access_token": "****",
  "refresh_token": "****",
  "expires_in": "300" ,
  "scope": "trapi.alerts.news.crud .... trapi.streaming.pricing.read",
  "token_type": "Bearer"
}
```

The response body is a JSON message. At this point the Refresh token should be stored for future use.

### Using the token

The Access token which we just received can now be used in the REST API data calls. Typically, this token is sent as a *Bearer* in the HTTP *Authorization* header in the request message. If the user tries to invoke an API call using an expired token, the server will reject the call. Here is an example of a data request and a rejected response:

```
GET http://api.refinitiv.com/data/historical-pricing/v1/views/interday-summaries/TRI.N HTTP/1.1
```

```
Host: api.refinitiv.com
```

```
User-Agent: python-requests/2.12.4
```

```
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Authorization: Bearer ****
```

Response:

**HTTP/1.1 401 Unauthorized**

```
Date: Wed, 18 Dec 2019 14:45:27 GMT
Content-Type: application/json
Content-Length: 119
Connection: keep-alive
Access-Control-Allow-Origin: *
www-Authenticate: Bearer realm="GET /data/historical-pricing/v1/views/interday-summaries/{universe}", scope="trapi.data.historical-pricing.summaries.read", error="invalid_token", error_description="token expired"
X-Tr-Requestid: 33603908-52b2-4e50-ba30-5336d116f50a
```

```
{
  "error": {
    "id": "33603908-52b2-4e50-ba30-5336d116f50a",
    "code": "401",
    "message": "token expired",
    "status": "Unauthorized"
  }
}
```

The server rejected our data API call with the "token expired" message. Typically, an application would keep track of when an Access token is about to expire and renew it before invoking a data call.

### Refreshing the token

Since the previous token is no longer valid, the user/application must get a new Access token, using the *Refresh Grant*:

```
POST http://api.refinitiv.com/auth/oauth2/v1/token
```

```
Host: api.refinitiv.com
User-Agent: python-requests/2.12.4
Accept-Encoding: gzip, deflate
Accept: application/json
Connection: keep-alive
Content-Length: 106
Content-Type: application/x-www-form-urlencoded
```

```
Authorization: Basic
RDZENZ*****DNkNEREQ3ND0=refresh_token=****&username=****&grant_type=refresh_to
ken
```

Response:

```
HTTP/1.1 200 OK
```

```
Date: Thu, 05 Apr 2018 19:24:02 GMT
```

```
Content-Type: application/json
```

```
Content-Length: 1604
```

```
Connection: keep-alive
```

```
X-Tr-Requestid: 4cbb7aaa-9492-499c-82f0-efd0f07bbf14
```

```
{
  "access_token": "****",
  "refresh_token": "****",
  "expires_in": "300" ,
  "scope": "trapi.alerts.news.crud .... trapi.streaming.pricing.read",
  "token_type": "Bearer"
}
```

## Token Scopes

Every resource (data REST API) within RDP has a permission set assigned to it. This permission set is called *scope*. The Access token must include the scope of the resource it is trying to access, otherwise the request will be rejected by platform.

For e.g. a News content REST API has a scope of **trapi.data.news.read**. The Access token must include **trapi.data.news.read** in addition to other scopes, to access news content data. This scope limiting is done when requesting the original token using Password Grant. The request parameter *Scope* can contain either a root scope or a subset. Requesting the top-level scope **trapi** will get us a token, which contains all the scopes we are entitled to.

Most applications will request a token with root scope: **trapi**. Following are the list of scopes defined within the RDP News and Alerts API, and may optionally be used when requesting the token:

- **trapi.alerts.news.crud**
- **trapi.data.news.read**

## 4. Request/Response Interface

The News dataset comprises of News Headlines and Stories. An application can use News Request-Response REST API to get the last few headlines. A link in the Headline points to a News Story, which can also be requested through a REST API call. Large Stories are paginated (broken up) across multiple requests.

### Base URI

The base URI for the News APIs is `<Environment URI>/data/news/v1`.

### Paths and Operations

The operations currently available on the News for RDP Request/Response interface are as follows:

| Path                       | Operation | Description                                                 |
|----------------------------|-----------|-------------------------------------------------------------|
| /analyze                   | GET       | Computes a tree expression from a user query string.        |
| /headlines                 | GET       | Snapshot of Headlines.                                      |
| /images/{imageid}          | GET       | Retrieves an image.                                         |
| /metadata                  | GET       | Retrieves the list of rcs code.                             |
| /metadata/{id}             | GET       | Retrieves newsCode metadata by id or by the readable value. |
| /metadata/{id}/children    | GET       | Retrieves the list of token children.                       |
| /online-reports            | GET       | Retrieves online reports hierarchy.                         |
| /online-reports/{reportid} | GET       | Retrieves online reports by id.                             |
| /readership/headlines      | GET       | Retrieves most read Headlines for a query.                  |
| /stories/{storyid}         | GET       | Retrieves a Story content.                                  |
| /suggest                   | GET       | Retrieves the token completion for a given query.           |
| /top-news                  | GET       | Retrieves the top news packages hierarchy.                  |
| /top-news/{topNewsId}      | GET       | Retrieves the top news headlines package.                   |

### Responses

A HTTP response status code is returned for the indicated HTTP request. The descriptions for each response code are listed below.

| Code | Response Description                                                                                                                                                                                                                                                               |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 200  | <b>Success</b><br>The request was successfully received, understood, and accepted. Matching bucket returned.                                                                                                                                                                       |
| 400  | <b>Bad Request</b><br>Can be the result of malformed request syntax, invalid request message parameters, or deceptive request routing, etc.                                                                                                                                        |
| 403  | <b>Forbidden</b><br>The request is formed correctly, but you do not have the necessary permissions for the resource.                                                                                                                                                               |
| 410  | <b>Gone</b><br>The requested resource has been permanently removed. It is similar to a 404 (Not found) code but is sometimes used in the place of a 404 for resources that used to exist but no longer do.                                                                         |
| 500  | <b>Internal Server Error</b><br>A generic error message indicating that an unexpected condition was encountered on the server side and no more specific message is suitable.<br>Returned most often when a web framework executes a request handler code that raises an exception. |

## Analyze operation

Analyze operation computes a tree expression from the user query string. It would break down the user query string into News related entities, then aggregate them into a Boolean filtering tree.

It helps to both understand the underlined concepts of the user query, then how they intricate together to filter news.

| Entities                      | Search query                                                               |
|-------------------------------|----------------------------------------------------------------------------|
| AND                           | <b>AND</b> operator indicates that all condition should be met             |
| OR operator                   | <b>OR</b> operator indicates that only one of the conditions could be met  |
| NOT                           | Condition inversion                                                        |
| Language mnemonic             | <b>LFR</b> (French)                                                        |
| Language RCS qCode            | <b>L:EN</b> (English)                                                      |
| RCS mnemonic code             | ODD                                                                        |
| Geography code                | UK                                                                         |
| Explicit token                | <b>NS</b> : NewsSource                                                     |
| Explicit RIC code             | R:MSFT.O                                                                   |
| Instrument code               | 0#.FTSE                                                                    |
| Explicit free text            | Use commas for explicit free text (for example: "USA")                     |
| Fretext destination modifiers | FullStory, HeadlineOnly                                                    |
| DateRange                     | Last X days, Before, Until, After, Since, Between, In, daterange:'from,to' |

## Resolving ambiguity, usage of prefixes

In general, the API attempts a best guess to resolve the nature of an item of the query. While it is obvious that MSFT.O is an equity and that LFR is a language, there could be ambiguity on some other examples:

| User Input | RCS Code                                      | RIC Code                     |
|------------|-----------------------------------------------|------------------------------|
| GS         | G:1N South Georgia and South Sandwich Islands | GS.N Goldman Sachs Group Inc |
| SIG        | M:2B Significant Company News                 | SIG.N Signet Jewelers Ltd    |

In those instances, where an input could be either an RCS Code or a RIC Code, the API will prioritize the RCS interpretation. GS will be seen as the RCS Code for the Islands, and SIG will be the RCS Code for Significant News.

The user can decide to explicitly raise the ambiguity and inform the API of the nature of the input by using prefixes. For instance, using the prefix `R:` will instruct the API that the code we reference is a RIC Code. Hence using `R:SIG` or `R:GS` will be recognized as Signet Jewelers and Goldman Sachs.

You will find in the following sections various prefixes that can be used to raise ambiguity of input items.

## Company prefixes

| Code Type     | Example      | Details                                      |
|---------------|--------------|----------------------------------------------|
| PermID        | P:4297144968 | Space Exploration Technologies Corp (SpaceX) |
| PermID        | P:4296373211 | Waitrose Ltd                                 |
| RIC           | R:TRI.TO     | Thomson Reuters                              |
| RIC           | R:MSFT.O     | Microsoft CP                                 |
| Secondary RIC | R:MSFT.OQ    | Microsoft CP                                 |

## Instruments

| Instrument type | Example   | Details               |
|-----------------|-----------|-----------------------|
| FX              | R:EUR=    | Euro                  |
| Commodity       | R:XAG=    | Silver                |
| Index           | R:.FTSE   | FTSE 100 Index        |
| Chain Index     | R:0#.FTSE | FTSE 100 constituents |

## RCS News Code

The general catch-all prefix for RCS code is `Topic:..`. Moreover, all search queries with the following prefixes will be considered as News Codes: U, E, G, A, B, I, M, F, C, BJ

| Code type        | Explicit query | Mnemonic query | Explanation                        |
|------------------|----------------|----------------|------------------------------------|
| Business sectors | B:128          | BNKS           | Banks                              |
| Business sectors | B:278          | TECH           | Technology                         |
| Languages        | L:EN           | LEN            | English                            |
| Languages        | L:FR           | LFR            | French                             |
| Geography        | G:A            | EUROP          | Europe                             |
| Geography        | G:6J           | US             | United States                      |
| Geography        | G:AM           | G7C            | G7 Countries                       |
| Others           | M:1S           | ODD            | Human Interest / Brights/ Odd News |
| Others           | M:1L           | NEWS1          | Major News                         |
| Others           | M:1QC          | GLOFIN         | Global Financial News              |
| Others           | M:1R7          | POTUS          | US President                       |

**Explicit query** and **Mnemonic query** are equivalent, so any convenient spelling is allowed. We recommend using explicit queries because mnemonic could be changed in the future.

## Source and Products

| Code type | Search query | Explanation                            |
|-----------|--------------|----------------------------------------|
| Source    | NS:RTRS      | Reuters News                           |
| Source    | NS:PLTS      | Platts                                 |
| Product   | NP:PGA       | Platts News Service - Global Alert     |
| Product   | NP:E         | Securities International News Service  |
| Product   | NP:C         | Commodities International News Service |

## Free text search

To search by free text without parsing to any symbols, companies, etc. use commas. For example:

- "United Kingdom"
- "Chocolate"
- "LEN" - search for free text 'LEN', not English. The result query can be the following: **LEN and "LEN"**. The result will contain all headlines in English with 'LEN' text in the story content or headline text.

### Substitution symbols

1. '?' matches one symbol

For example:

- **US?** matched **USA, USB**
- **US?R** matched **USER, USSR**

2. '\*' matches any number of symbols

For example:

- **US\*** matches **USA, US, USABLE**
- **US\*E** matched **USABLE**

### SearchIn token

SearchIn token can be used only with FreeText search. If it is specified for non-freetext search (for example, with prefix) SearchIn token will be ignored.

| Token        | Search query example            | Explanation                                                 |
|--------------|---------------------------------|-------------------------------------------------------------|
| FullStory    | OBAMA AND SearchIn:FullStory    | Search for <b>OBAMA</b> in both headline and story contents |
| HeadlineOnly | OBAMA AND SearchIn:HeadlineOnly | Search for <b>OBAMA</b> in headline text only               |

## Data Range

There are two supported ways of setting a date range. First, you can use the dateFrom/dateTo query string parameters, which supports the ISO-8601 format **YYYY-MM-DDTHH:mm:ss.SSSZ**, examples: "2011-10-05T14:48:56.678Z", "2015-12-22T16:48:34.345Z". Else, you can also set the date range as part of the query, this second option takes the precedence before the query string parameters, and supports a variety of human friendly syntax:

## Boolean Range

### OR usage

As a shorthand, the usage of a comma is interpreted as a OR operator.

| Search query example                 | Parsed Query                                                 | Executed Query                  | Explanation                                                           |
|--------------------------------------|--------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------|
| L:EN <b>or</b> L:FR                  | Language:LEN <b>OR</b> Language:LFR                          | L:en OR L:fr                    | English or French using an explicit OR operator                       |
| L:EN , L:FR                          | Language:LEN <b>OR</b> Language:LFR                          | L:en OR L:fr                    | English or French using a comma in place of a OR                      |
| EURO/COAL <b>OR</b> CO2              | Report:EURO/COAL <b>OR</b> Topic:CO2                         | RR:1141 OR U:E                  | European daily physical coal report or Carbon/Emissions Markets       |
| POWER/DE <b>OR</b> POWER/FR and l:en | (Report:POWER/DE <b>OR</b> Report:POWER/FR) AND Language:LEN | ( RR:2925 OR RR:2927 ) AND L:en | (Power outage diary Germany or Power outage diary France) and English |
| VEN AND (NGS <b>OR</b> WEA)          | Topic:VEN AND (Topic:NGS <b>OR</b> Topic:WEA)                | M:MQ AND ( U:2R OR U:I )        | Service Alerts and (Natural Gas or Weather Markets/Weather)           |

### AND usage

| Search query example                                 | Parsed Query                                  | Executed Query                  | Explanation                                              |
|------------------------------------------------------|-----------------------------------------------|---------------------------------|----------------------------------------------------------|
| NGS <b>and</b> l:en <b>and</b> EUROP                 | Topic:NGS AND Language:LEN AND Topic:EUROP    | U:2R AND L:en AND G:A           | Natural Gas and English and Europe                       |
| l:en <b>and</b> RNW <b>AND</b> EUROP                 | Language:LEN AND Topic:RNW AND Topic:EUROP    | L:en AND U:N AND G:A            | English and Renewable Energy and Europe                  |
| EL/CEE <b>and</b> l:en                               | Report:EL/CEE AND Language:LEN                | RR:1864 AND L:en                | Central European power market and English                |
| M:U <b>AND</b> NS:RTRS <b>AND</b> G:AE <b>AND</b> AA | Topic:OLY AND Source:RTRS AND Topic:KR AND AA | M:U AND NS:RTRS AND G:AE AND AA | Olympics and Reuters News and South Korea and All Alerts |

## NOT usage

| Search query example       | Parsed Query                             | Executed Query            | Explanation                                                                                 |
|----------------------------|------------------------------------------|---------------------------|---------------------------------------------------------------------------------------------|
| (REMIT AND PWR) AND NOT GB | Product:REMIT AND Topic:PWR NOT Topic:GB | NP:REMIT AND U:G NOT G:7J | Regulation on Energy Market Integrity and Transparency and Power Markets NOT United Kingdom |
| COA NOT COAL/TEND          | Topic:COA NOT Report:COAL/TEND           | U:H NOT RR:1221           | Coal NOT News on coal market tenders                                                        |
| U:47 AND NS:RTRS NOT AA    | Topic:POIL AND Source:RTRS NOT AA        | U:47 AND NS:RTRS NOT AA   | Palm Oil and Reuters News NOT All Alerts                                                    |

## Complex Boolean usage and parenthesis

The three kind of Boolean operator can be combined to build any sort of more complex query. The service attempts to do a best guess as to what would be the most natural priority. In order to raise any ambiguity, parenthesis can be used.

| Search query example                              | Executed Query                                                         | Explanation                                                                                             |
|---------------------------------------------------|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| L:EN OR L:FR AND R:TRI.TO                         | ( Language:LEN OR Language:LFR ) AND R:TRI.TO                          | The service assumes that the two languages should be grouped together first, then combined with the RIC |
| L:EN OR ( L:FR AND R:TRI.TO )                     | Language:LEN OR ( Language:LFR AND R:TRI.TO )                          | We used parenthesis to enforce the grouping of LFR and TRI.TO together                                  |
| R:INTC.O OR R:TRI.TO AND L:EN NOT WEA             | ( R:INTC.O OR R:TRI.TO ) AND Language:LEN AND ( NOT ( Topic:WEA ) )    | RICs are grouped together, also the service infers the usage of a AND before the NOT                    |
| L:EN AND (INTC.O OR MSFT.O OR (RTRS AND TRI.TO) ) | Language:LEN AND (R:INTC.O OR R:MSFT.O OR (Source:RTRS AND R:TRI.TO) ) | You can combine parenthesis to reach any depth and complexity                                           |

You can call the /analyze endpoint to see how the priority was inferred by the service.

## Common and useful codes

Some codes can be useful to search for:

- `M:1RS` for most read news.
- `Topic:SIG` for significant company news.

## Headlines operation

Get headlines operation consists of two parts:

1. Analyze search query.
2. Retrieve data for parsed query.

It means that all rules mentioned in [Analyze operation](#) section can be used for the headlines search query.

## Request options

### Query

The user search query. Refer to [Analyze operation](#) section for more details.

### Limit

Limit of the headline (default 10, range value: [0, 100]).

### Cursor

The pagination cursor. Refer to [Headlines Pagination](#) section for more details.

### Sort Order

Support three values:

- New to Old (default)
- Old to New
- Most Read

If **MostRead** sort order selected, you can specify date range for most read headlines using **mostReadDateFrom** and **mostReadDateTo** parameters.

For example: /headlines?query=TOPNEWS&sort=mostRead&mostReadDateFrom=2011-10-10&mostReadDateTo=2018-10-10

If **mostReadDateFrom** or/and **mostReadDateTo** specified but the sort order is not equal to **mostRead**, the values will be ignored.

### Relevancy

Three values are supported:

- Filter on "High" relevance should return the results for just HIGH
- Filter on "Medium" relevance should return the results for Medium+HIGH
- Filter on "All" relevance should return the results for Low+Medium+HIGH

## Response in details

For example, if searching for 'POTUS' (President of the United States), we get the response with the following fields:

### Story ID

Can be used to retrieve full story content using [Stories operation](#).

```
{
  "storyId": "urn:newsml:reuters.com:20180228:nNRA6d1exq:1"
}
```

### Version

**On each news update**(revision) **version** became increased.

```
"_version": 1
```

### Creator

Creator of the news. In the example below **NS:INDEPE** refers to "The Independent" newspaper.

```
"creator": [
  {
    "_qcode": "NS:INDEPE",
    "_role": "sRole:source"
  }
]
```

### Info Source

Source of the news. In the example below **NS:INDEPE** refer to "The Independent" newspaper.

```
"infoSource": [
  {
    "_qcode": "NS:INDEPE",
    "_role": "sRole:source"
  },
  {
    "_qcode": "NS:INDEPE",
```

```

        "_role": "sRole:origProv"
      }
    ]

```

### Language

```

"language": [
  {
    "_tag": "en"
  }
]

```

### Subject

**Array** contains **all** keywords related **with** headline.

In example below we have the following subject-related keywords:

- **G:1** refers to **Emerging Market Countries**
- **G:2V** refers to **Mexico**
- **G:4** refers to **Americas**
- **M:1QD** refers to **Suggested Sources**
- **M:4** refers to **Asylum / Immigration / Refugees**
- **M:A** refers to **Cryme**

```

"subject": [
  {
    "_qcode": "G:1"
  },
  {
    "_qcode": "G:2V"
  },
  {
    "_qcode": "G:4"
  },
  {
    "_qcode": "M:1QD"
  },
  {
    "_qcode": "M:4"
  },
]

```

```
{
  "_qcode": "M:A"
}
```

#### Urgency

Urgency of the news. Can be **Hot** or **Regular**.

```
"urgency": {
  "$": 3
}
```

#### News creation date

```
"firstCreated": {
  "$": "2018-02-28T08:35:17.006Z"
}
```

#### News update date

```
"versionCreated": {
  "$": "2018-02-28T08:35:17.006Z"
}
```

#### Headline title

Short description of the news.

```
"title": [
  {
    "$": "Trump administration can waive environmental rules over Mexico border wall, judge rules During the presidential campaign, Donald Trump accused Judge Gonzalo Curiel of being 'very hostile' and cited his Mexican heritage"
  }
]
```

#### Headlines pagination

Headline pagination is cursor based.

## Images operation

Stories which are part of a Top News or an Online Report package may have an image associated with. It will be presented in the form of an identifier you can use in the **/images/{id}** endpoint.

There you can retrieve an image either as a raw binary, or as a JSON RSF document.

When loading as a raw binary, you can also create a thumbnail for the image.

### Basic usage

The only mandatory field is the image identifier.

Depending on the source of the image, Online Report or Top News package, the format will be slightly different.

The image id field will then be part of the URL path: **/images/{id}**

For instance, with ID 58c94894-f0c9-49ca-80c9-69b9bd099f7f the resolved path is:

```
/images/58c94894-f0c9-49ca-80c9-69b9bd099f7f
```

### Supported output formats

You can switch between the raw image representation and a JSON RSF output using the accept header.

There are two possible values:

- **image/jpeg** - the default, provides the raw image output
- **application/json** - provides the image in a JSON RSF format, which allows for additional metadata about the image to be retrieved

### Thumbnail generation

When loading the image using the **image/jpeg** format, you can create thumbnail on the fly using the **height** and **width** headers.

If you specify only one of the two sizes, the services computed the other sizes to keep the image original ratio.

When the two sizes are defined and ratio is changed, the image is stretched and cropped to fill the whole output.

## Metadata operations

Metadata endpoints allow to navigate the list of News codes. Those codes are layered into a hierarchy, the root code being M:1.

Parts of the hierarchy are Language codes, another Business Sector codes, another Geography codes, etc.

This hierarchy is mostly a tree, but in some places an item can belong to several sections, and thus have more than one parent.

Some codes exist only to group other codes in the hierarchy, in other words not all the codes are searchable and can give result. You can see that by checking the searchable field.

There are more than 20 000 codes.

## Available operations

We provide 3 main operations:

- **/metadata** allows to dump a flat list of all the News codes
- **/metadata/{id}** gives the full description of a known code specified by its **id** or **readable**
- **/metadata/{id}/children** gives the list of children of a known code specified by its **id** or **readable**, this allows to explore the hierarchy, the root being **M:1**

## Metadata fields

All the codes have the following fields:

- **id** - the unique ID of the code.
- **readable** - a more human friendly identifier for the code, thus it is subject of changing through time, we recommend storing and using the id.
- **label** - the label of the code.
- **description** - a full description of the code, not always available.
- **group** - the subgroup to which the code belongs.
- **searchable** - whether the code is searchable and can give results.
- **children** - the number of children the code has.

Example of news code:

```
{
  "newsCode": {
    "id": "B:227",
    "description": "Manufacturers, extractors and refiners of chemicals, minerals,
precious metals, steel, aluminum, forest products, and construction and other raw
materials.",
    "label": "Basic Materials (TRBC level 1)",
    "group": "BusinessSectors",
    "readable": "Topic:BMAT",
    "searchable": true,
    "childrenCount": 105
  }
}
```

## Group Filter

Available for **/metadata** endpoint only.

Allows to perform filtering by group.

Supported groups:

- BusinessSectors
- Geography
- Language
- Market
- MoreTopics
- Report
- Source

Example:

```
/metadata?group=Source
```

### accept-language header

accept-language header allows to select response language.

Not all the exposed metadata have translation. If a non-English label is not available, we fallback to English.

Supported languages:

- en - English
- zh - Chinese
- ja - Japanese
- ko - Korean

We do support the language header to be sent in the RFC 7231 format:

<https://tools.ietf.org/html/rfc7231#section-5.3.5>

For instance, these are valid inputs:

```
en
```

```
en-US
```

```
en-GB,en;q=0.8
```

If no accept-language header specified or specified any non-supported value language will be defaulted to English (**en**).

### Metadata pagination

We use two different pagination schemes in the metadata endpoints.

- Cursor pagination is used for **/metadata**
- Offset pagination is used for **/metadata/{id}/children**

You can always control the size of the page you want to paginate using the limit query string parameter.

```
/metadata?limit=20
```

## Online Reports

### Hierarchy of online reports

No input required. Response will be the hierarchy of online reports in JSON format.

Example of response:

```
{
  "data": [
    {
      "name": "Japan Online Report",
      "reports": [
        {
          "reportId": "OLJPENT",
          "description": "Entertainment News",
          "language": "Japanese"
        },
        {
          "reportId": "OLJPODD",
          "description": "Oddly Enough",
          "language": "Japanese"
        }
      ]
    },
    {
      "name": "Brazil Online Report",
      "reports": [
        {
          "reportId": "OLBRDOM",
          "description": "Domestic News",
          "language": "Portuguese"
        }
      ]
    }
  ]
}
```

### Curated report

The search value should correspond to the report of interest, e.g. OLGBTOPNEWS. Response will be report in JSON format.

## Full Document Top 10

The search value should correspond to the report of interest, e.g. OLGBTOPNEWS. The `fullContent` parameter should be set to `true`. The response will be the top 10 full documents for report in JSON format.

## Images

The Curated Report and Full Document Top 10 will contain image links. An image ID is composed of a GUID and Rendition, separated by a forward slash character. Example: **5d2ff627-6683-467c-abc-b-7ac908883bb9/rend:baseImage**. Use GUID identifier only for the image retrieval. The response will be a base 64 encoded image or story in JSON format with encoded image in it. Refer to [Images operation](#) for more information.

## Stories operation

Gets story by specific full story identifier

- **storyId**: full story identifier (for example: "urn:newsml:reuters.com:20180227:nNRA6cw73t:1")

## HTML and JSON views

Story can be retrieved in JSON or HTML format.

To select story format 2 options are available:

- **accept** header:
  - text/html
  - application/json
- extension
  - json
  - html

Extension has the higher priority than the header. It means that if, for example, the accept header equals to "text/html", but story ID is sent as **sampleStory.json**, the result will be in JSON format.

### HTML view response

| Tag   | Explanation                              | Example                                                                                           |
|-------|------------------------------------------|---------------------------------------------------------------------------------------------------|
| head  | Container for metadata (data about data) | <head><title>GRAPHIC-Italy's uncertain election: Five questions on investors'minds</title></head> |
| title | Defines the title of the document        | <title>GRAPHIC-Italy's uncertain election: Five questions on investors' minds</title>             |

HTML view response is a response with 307 status - Temporary redirect. Redirect location can be found in **location** header. In addition to the content, HTML view response can also contain regular HTML tags for formatting. The part of the content which contains the headline can contain tags.

Part of the story where the content itself is situated can contain:

| Tag   | Explanation                     | Example                                                                                                                                                                     |
|-------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| p     | Paragraph                       | <code>&lt;p class="tr-summaryitem"&gt;Polls point to hung parliament in Italy&lt;/p&gt;</code>                                                                              |
| ul    | Unordered list                  | <code>&lt;ul&gt;&lt;li&gt;AlfaSigma SpA&lt;/li&gt;&lt;/ul&gt;</code>                                                                                                        |
| ol    | Ordered list                    | <code>&lt;ol&gt;&lt;li&gt;AlfaSigma SpA&lt;/li&gt;&lt;/ol&gt;</code>                                                                                                        |
| li    | List item                       | <code>&lt;li&gt;Galectin Therapeutics Inc&lt;/li&gt;</code>                                                                                                                 |
| table | For creating table              | <code>&lt;table cellpadding="3" cellspacing="0" class="tr-social-content"&gt;</code>                                                                                        |
| tr    | Table row                       | <code>&lt;tr&gt;&lt;td&gt;&lt;img height="48" src="http://pbs.twimg.com/profile_images/968668827411206145/uXb5SXTc_normal.jpg" width="48"/&gt;&lt;/td&gt;&lt;/tr&gt;</code> |
| td    | Table data/cell                 | <code>&lt;td&gt;&lt;img height="48" src="http://pbs.twimg.com/profile_images/968668827411206145/uXb5SXTc_normal.jpg" width="48"/&gt;&lt;/td&gt;</code>                      |
| a     | For creating links              | <code>&lt;a href="https://twitter.com/NadaanMundaa"&gt;Nadaan Munda @NadaanMundaa&lt;/a&gt;</code>                                                                          |
| b     | Specifies bold text             | <code>&lt;b&gt;What is Reuters News Tracer?&lt;/b&gt;</code>                                                                                                                |
| img   | Generic inline container        | <code>&lt;img height="48" src="http://pbs.twimg.com/profile_images/968668827411206145/uXb5SXTc_normal.jpg" width="48"/&gt;</code>                                           |
| span  | Defines a section in a document | <code>&lt;span style="float: left; width: 50%; text-align: left"&gt;Part of the text to highlight or mark with additional style(class)&lt;/span&gt;</code>                  |
| div   | Division or a section           | <code>&lt;div class="body-dark"&gt;</code>                                                                                                                                  |
| br    | Line break                      | <code>&lt;b&gt;What is Reuters News Tracer?&lt;/b&gt;&lt;br/&gt;</code>                                                                                                     |

Also, CSS classes can be used inside HTML tags for additional formatting, for example:

| Class             | Explanation                                                                                                     | Example                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| tr-story-pl       | Identifies first story paragraph                                                                                | <code>&lt;p class="tr-story-pl"&gt;</code>                                                                                |
| tr-pnac           | Placeholder for links to full stories                                                                           | <code>&lt;span class="tr-pnac" id="x1"&gt;nL8N1PB47V&lt;/span&gt;</code>                                                  |
| tr-ric            | Placeholder for RIC                                                                                             | <code>&lt;span class="tr-ric"&gt;[RIC]&lt;/span&gt;</code>                                                                |
| tr-link           | Placeholder for links                                                                                           | <code>&lt;a class="tr-link" href="http://reut.rs/2BEp89X"&gt;http://reut.rs/2BEp89X&lt;/a&gt;</code>                      |
| tr-resource-ref   | Placeholder for images                                                                                          | <code>&lt;span class="tr-resource-ref" id="d0f3357e-667e-4ele-bde0-a8b35a894c74"&gt;&lt;/span&gt;</code>                  |
| tr-sandwichbox    | Placeholder for containing related information (links, story or graphic)                                        | <code>&lt;div class="tr-sandwichbox"&gt;</code>                                                                           |
| tr-link-email     | Placeholder for mailto-links                                                                                    | <code>&lt;a class="tr-link tr-link-email" href="mailto:terence.gabriel@tr.com"&gt;terence.gabriel@tr.com&lt;/a&gt;</code> |
| tr-summaryinfo    | Placeholder for creating summary                                                                                | <code>&lt;div class="tr-summaryinfo"&gt;</code>                                                                           |
| tr-summaryitem    | Items of summary in placeholder                                                                                 | <code>&lt;p class="tr-summaryitem"&gt;Polls point to hung parliament in Italy&lt;/p&gt;</code>                            |
| tr-dateline       | Placeholder for dates                                                                                           | <code>&lt;span class="tr-dateline"&gt;LONDON, Feb 26 (Reuters)&lt;/span&gt;</code>                                        |
| tr-dl-sep         | Placeholder for dates separator. Always "-"                                                                     | <code>&lt;span class="tr-dl-sep"&gt;[dateline separator]&lt;/span&gt;</code>                                              |
| tr-signoff        | Placeholder with details about story owner/editor/creator                                                       | <code>&lt;p class="tr-signoff"&gt;</code>                                                                                 |
| tr-contactinfo    | Placeholder for contacts section                                                                                | <code>&lt;p class="tr-contactinfo"&gt;</code>                                                                             |
| tr-social-content | Placeholder, which contains links to social networks                                                            | <code>&lt;table cellpadding="3" cellspacing="0" class="tr-social-content"&gt;</code>                                      |
| tr-advisory       | Placeholder for information must be entered in the advisory field                                               | <code>&lt;p class="tr-advisory"&gt;Adds Chinese comment&lt;/p&gt;</code>                                                  |
| tr-additinfo      | Information to be entered in the additional information field                                                   | <code>&lt;div class="tr-additinfo"&gt;[contents of the additional information field]&lt;/div&gt;</code>                   |
| tr-by             | Information without conjunction words like words "by" and "and", as this are added automatically on publication | <code>&lt;p class="tr-by"&gt;By Steve Holland and Frank Jack Daniel&lt;/p&gt;</code>                                      |
| tr-em             | Placeholder for italic text                                                                                     | <code>&lt;span class="tr-em"&gt;Contacts&lt;/span&gt;</code>                                                              |

|                     |                                          |                                                                                                           |
|---------------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| tr-legacyusage-news | Placeholder for unidentified news entity | <code>&lt;span class="tr-legacy-usage-news"&gt;[unidentified item in square brackets]&lt;/span&gt;</code> |
| tr-ni               | Placeholder for valid RCS code           | <code>&lt;span class="tr-ni"&gt;[Named item code]&lt;/span&gt;</code>                                     |
| tr-underline        | Placeholder for underlined text          | <code>&lt;span class="tr-underline"&gt;Berlin&lt;/span&gt;</code>                                         |

## NewsWire response in details

### Story content

**inlineData** and **inlineXml** is a different representation of the same data. inlineData is a text representation, inlineXml is a formatted HTML markup which is ready to be used on HTML pages.

```
"contentSet": {
  "inlineData": {
    "$": "string",
    "_contentType": "string"
  },
  "inlineXml": {
    "$": "string",
    "_contentType": "string"
  }
}
```

### Creator

Creator of the news. In the example below **NS:INDEPE** refer to "The Independent" newspaper.

```
"creator": [
  {
    "_qcode": "NS:INDEPE",
    "_role": "sRole:source"
  }
]
```

### Urgency

Urgency of the news. Can be **Hot** or **Regular**.

```
"urgency": {
  "$": 3
}
```

### Info Source

Source of the news. In the example below **NS:INDEPE** refer to "The Independent" newspaper.

```
"infoSource": [
  {
    "_qcode": "NS:INDEPE",
    "_role": "sRole:source"
  },
]
```

```

    {
      "_qcode": "NS:INDEPE",
      "_role": "sRole:origProv"
    }
  ]

```

#### Language

```

"language": [
  {
    "_tag": "en"
  }
]

```

#### Subject

**Array** contains all keywords related with headline.

In example below we have the following subject-related keywords:

- **G:1** refers to **Emerging Market Countries**
- **G:2V** refers to **Mexico**
- **G:4** refers to **Americas**
- **M:1QD** refers to **Suggested Sources**
- **M:4** refers to **Asylum / Immigration / Refugees**
- **M:A** refers to **Crime**

```

"subject": [
  {
    "_qcode": "G:1"
  },
  {
    "_qcode": "G:2V"
  },
  {
    "_qcode": "G:4"
  },
  {
    "_qcode": "M:1QD"
  },
  {
    "_qcode": "M:4"
  }
]

```

```

    },
    {
      "_qcode": "M:A"
    }
  ]

```

#### News update date

```

"versionCreated": {
  "$": "2018-02-28T08:35:17.006Z"
}

```

#### News creation date

```

"firstCreated": {
  "$": "2018-02-28T08:35:17.006Z"
}

```

#### Title

Short description of the news.

```

"title": [
  {
    "$": "Bridgestone Enjoys Successful Olympic Winter Games Debut, Turns
Attention to Olympic Games Tokyo 2020; worldwide Olympic Partner&apos;s first global
pro..."
  }
]

```

#### Clustering Keys

Some stories could contain Clustering (deduplication) keys.

Dedupe Key Type (dedupeKeyType property):

| Deduplication Key Type | Description                                  | Example                                      |
|------------------------|----------------------------------------------|----------------------------------------------|
| canna                  | Clustering by Associative News NLP Algorithm | 20190710-101241                              |
| eaId                   | Editorial Asset Id                           | 217f6e27-1ea3-e911-e49a-215056933aa7         |
| strict                 | Strict                                       | -<br>181af4a8c55579fe576b7b81efd839b8095eeb5 |

```

"contentMetaExtProperty": [
  {
    "_creator": "rftResRef:sys24",
    "_how": "howextr:tool",
    "_id": "s1m220",
    "_qcode": "dedupeKey:canna:20190710-106286",
    "_rel": "extCptRel:hasDedupeKey",
    "_why": "why:inferred",
    "related": [
      {
        "_creator": "rftResRef:sys24",
        "_how": "howextr:tool",
        "_qcode": "dedupeKeyType:canna",
        "_rel": "extCptRel:hasDedupeKeyType",
        "_why": "why:inferred"
      }
    ]
  }
]

```

## Suggest operation

Suggest endpoint provides tokens completion for the user query. For EDP, the results come only from RCS knowledge base, which is stored in Redis database or AWS DynamoDB.

The cache strategy is applied. Values are cached for 24 hours.

## Suggest Parameters

### Query

The user query, which can be used in the suggestion features.

### Items per group

Suggestion results could be from different groups, like Geographies, Business Sectors, Organization, Indices, etc.

`itemsPerGroup` parameter specifies number of items in those groups, not a total items number.

### accept-language header

`accept-language` header allows to select response language.

Supported languages:

- en - English

- zh - Chinese
- ja - Japanese
- ko - Korean

If no `accept-language` header is specified or any non-supported value is specified, the language will be defaulted to English (**en**).

## Top News operations

### Packages

No input required. The response will be the hierarchy of top news packages in JSON format.

Example of response:

```
{
  "data": [
    {
      "name": "Main",
      "pages": [
        {
          "name": "Front Page",
          "topNewsId": "urn:newsml:reuters.com:20020923:SPDOC_119827232002"
        },
        {
          "name": "Central Banks & Global Economy",
          "topNewsId": "urn:newsml:reuters.com:20080317:SPDOC_1727041720081"
        }
      ]
    },
    {
      "name": "Breakingviews",
      "pages": [
        {
          "name": "Reuters Breakingviews",
          "topNewsId": "urn:newsml:reuters.com:20090126:SPDOC_1343432620091"
        }
      ]
    }
  ]
}
```

### Headlines

Can be used to get top news headlines. Package ID is required.

Response example:

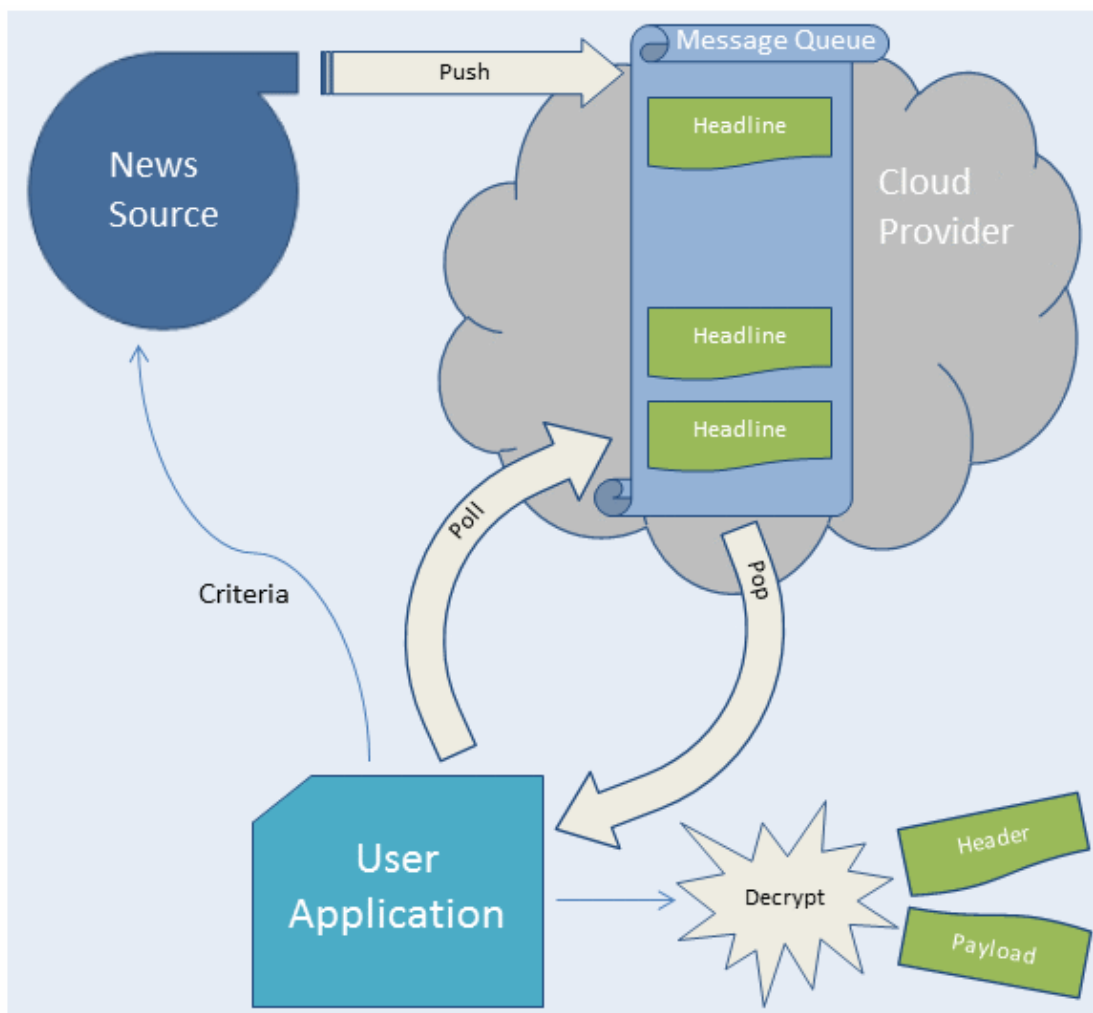
```
{
  "data": [
    {
      "text": "Soccer-Trophies more important for Man United than top four -
Solskjaer",
      "dateLine": "Jan 29 (Reuters)",
      "snippet": "Manchester United caretaker manager Ole Gunnar Solskjaer said he
will not be satisfied.",
      "image": {
        "byLine": "HANNAH MCKAY",
        "text": "A file photo of Manchester United interim manager.",
        "smallId": "2019-01-29T082009Z_1_CW0_RTRLXPP_1_LYNXPACKAGER.JPG",
        "id": "2019-01-29T082009Z_1_CW0_RTRLXPP_2_LYNXPACKAGER.JPG"
      },
      "versionCreated": "2019-01-29T05:23:00.000Z",
      "storyId": "urn:newsml:reuters.com:20190129:nL5N1ZT0GG",
      "relatedHeadlines": [
        {
          "text": "Soccer-Chelsea to face Manchester United in FA Cup fifth round",
          "versionCreated": "2019-01-28T19:50:00.000Z",
          "dateLine": "Jan 28 (Reuters)",
          "snippet": "Holders Chelsea have been drawn to face Manchester United.",
          "webUrl": "",
          "documentType": "story",
          "storyId": "urn:newsml:reuters.com:20190128:nL5N1ZS63V"
        }
      ]
    },
    ...
  ]
}
```

## 5. Alerts Delivery

The most typical use case for getting News Updates, is that an application starts a subscription, and then listens to the Updates as they happen - in an asynchronous manner. It could be for a news blotter on the web page, or a streaming headline display on a large wall board monitor. This is achieved through the Alerts delivery mechanism.

This scenario would comprise the following activities:

- The application starts by expressing an interest in Alerts to the news source.
- The news service creates a queue in cloud and starts pushing the messages that match the requested criteria. The message can be a News Headline or a Story depending on the request criteria.
- The application polls the queue for any available message. Awaiting messages are retrieved and removed from the queue.
- Individual messages are intended for the requesting application only and have to be decrypted to get the content payload. No other application can read the messages, even if it gets the access to the queue in the cloud.



## Workflow to subscribe to Alerts

Let's check the API calls for starting and subscribing to News Alerts. An application has to do following high level tasks:

1. Login.
2. Subscribe to the Alerts.
3. Get the cloud credentials.
4. Poll and retrieve the messages
5. Decrypt the messages.

### Login

Login is the first and common step for all RDP API requests. RDP supports OAuth 2.0 authentication with Password and Refresh grants.

For details about Login messages and implementation details, see the [Authentication](#) chapter.

### Alerts Subscription

RDP News Alerts (or Research Alerts) service is invoked with the request parameters which convey the delivery endpoint and type of Alerts to receive. An example of *breaking, English language* News for *Apple Inc* is:

```
{
  "transport": {
    "transportType": "AWS-SQS"
  },
  "filter": {
    "urgency": "1",
    "language": "en",
    "subjects": "R:AAPL.O"
  }
}
```

where, a user is asking for real time top-news, and intends to use Amazon's Simple Queue Service as the message distributor. Currently AWS-SQS is the only supported transport.

There are three news content subscriptions available currently:

- **news-headlines-subscriptions** – for Headline + metadata only.
- **news-stories-subscriptions** - for Headline + metadata + Story Body.
- **news-subscriptions** – provides one each of Headline and a Story, i.e. duplicate Updates.

On the RDP side, this request will:

1. Create a queue in AWS.
2. Start pumping the messages that meet the requested criteria into the queue.

3. Return the newly created queue parameters like:
  - a. `endpoint` - the URL of the queue that was created for us.
  - b. `cryptographyKey` - the Alert messages are encrypted before pushing into the queue. This key is required for decryption.
  - c. `subscriptionID` - a unique identifier, which will be used to close the subscription.

## Create the Credentials to a Cloud Queue

Applications need the access credentials to access the third-party cloud-based message queue. RDP, which is the owner of the queue, can generate these client credentials. In this step, the application invokes the RDP **cloud-credentials** service, and passes into URL the queue endpoint that was previously created.

RDP responds with `accessKeyId`, a `secretKey` and `sessionToken`, which together form the cloud credentials. These credentials expire every hour, so the application has to renew it periodically.

## Polling and Retrieving Messages

The cloud credentials will enable an application to perform inquiry and retrieval operations on the queue. It is now application's responsibility to:

1. Poll the queue endpoint to see if any messages are waiting.
2. Receive the messages, if available.
3. Delete the message which has been processed from the queue; to avoid getting duplicates.

Any message that is not retrieved from the queue, will automatically expire after 14 days, and be lost forever.

The semantics of message retrieval are specific to the cloud provider. For Amazon SQS Queue, please refer to this [documentation](#).

## Decrypting Messages

The messages in the queue are encrypted with credentials, which are only known to the application which created the request. This ensures that even if another application/party were to get hold of cloud-based queue, and retrieve messages from it, the payload would be undecipherable. The messages are first encrypted using AES256 with GCM, and then base64 encoded before being pushed into the queue.

The key parameters with which alerts message is encrypted are:

```
AES Key Length = 256 bytes
GCM AAD Length = 16 bytes
GCM TAG Length = 16 bytes
GCM NONCE Length = 12 bytes
```

To decrypt the message, application should:

1. Prepare the AES-256 secret key by base64 decoding the `cryptographyKey`.

2. Prepare the *cipher text bytes* by base64 decoding the message body.
3. Remove *Tag* from the *cipher text*.
4. Prepare *AAD* by copying the first 16 bytes from *cipher text*.
5. Prepare *NONCE* (IV) by copying bytes 4 – 16 from *AAD*.
6. Then decrypt with AES – GCM, tag length = 16 bytes; with the input from 1, 3, 4, 5.

Most programming languages provide Cryptography libraries, like `javax.crypto.*` in Java and `pycryptodonein` Python, to handle the underlying task of decoding and decryption.

## Message Payload

The decrypted message payload is a JSON data structure which contains all the information for that particular alert.

The JSON structure contains meta-data about the alert message, and either a key `payload` or `href`. The *payload* key will be used when all the information pertaining to the alert, can be included in the packed message. For very large payloads like a news story or company report, a file will be available for download in the cloud service, and href key will contain the `claim check` (URL) of this large payload. A *claim check* is typically valid for 14 days. The application should be designed to follow the URL in the key *href*, to retrieve the content.

Sample News structure with payload:

```
{
  "attributes": [{
    "domain": {
      "type": "string",
      "value": "headline"
    }
  ]},
  "envelopeVersion": "1.0",
  "ecpMessageID": "urn:newsml:reuters.com:20181016:nL8N1WW4V6:2",
  "sourceSeqNo": 2539219,
  "distributionSeqNo": 4,
  "sourceTimestamp": "2018-10-16T13:45:00.000Z",
  "receiveTimestamp": "2018-10-16T13:45:00.167Z",
  "distributionTimestamp": "2018-10-16T13:45:00.496Z",
  "payloadVersion": "1.0",
  "subscriptionID": "d61b1c7c-670f-419e-8d48-161f0ccba823",
  "payload": {
    "newsMessage": {
      "_xmlns:xs": "http://www.w3.org/2001/XMLSchema",
```

```

        " xsi:schemaLocation": "...
http://www.reuters.com/ns/2003/08/content ...",
        "_xmlns": "http://iptc.org/std/nar/2006-10-01/",
        .
        .
        .

```

Sample News structure with claim check:

```

{
  "attributes": [{
    "domain": {
      "type": "string",
      "value": "story"
    }
  ]},
  "envelopeVersion": "1.0",
  "ecpMessageID": "urn:newsml:reuters.com:20180718:nASX572h10:1",
  "sourceSeqNo": 150877,
  "distributionSeqNo": 127,
  "sourceTimestamp": "2018-07-18T08:15:49.236Z",
  "receiveTimestamp": "2018-07-18T08:15:51.263Z",
  "distributionTimestamp": "2018-07-18T09:29:36.169Z",
  "payloadVersion": "1.0",
  "subscriptionID": "b32c64f7-c82b-4d23-9055-257cdd1b3e51",
  "href": "https://***/file-store/files/3b5d1a9c-9eee-4928-9640-18e8a06b1d8a/stream...",
  .

```

## Output

Following is the sample output from an alerts service.

```

{
  "attributes": [{
    "domain": {
      "type": "string",
      "value": "headline"
    }
  ]},
  "envelopeVersion": "1.0",
  "ecpMessageID": "urn:newsml:reuters.com:20181113:nFWN1XO0WI:1",
  "sourceSeqNo": 408579,

```

```

"distributionSeqNo": 1,
.
.
.
"payload": {

    "itemClass": {
        "_rtr:msgType": "S",
        "_qcode": "ninat:text"
    },
    "versionCreated": {
        "$": "2018-11-13T19:18:47.25Z"
    },
    "generator": [{
        "$": "UCDP:parasn_alerts_rsf_10.54.4.67_1.2.39621:",
        "_versioninfo": "1.0.39621"
    }
    ],
    "title": [{
        "$": "LG OBERLIN, LLC FILES TO SAY IT HAS RAISED $22.2 MLN IN EQUITY
FINANCING - SEC FILING"
    }
    ],
.
.
}

```

## Base URI

The base URI for the News APIs is <Environment URI>/data/news/v1.

## Paths and Operations

The operations currently available on the News for RDP Request/Response interface are as follows:

| Path                          | Operation | Description                                                        |
|-------------------------------|-----------|--------------------------------------------------------------------|
| /news-headlines-subscriptions | GET       | Creates and manages notifications for News Headlines publications. |
|                               | POST      |                                                                    |
|                               | DELETE    |                                                                    |
| /news-stories-subscriptions   | GET       | Creates and manages notifications for News Stories publications.   |
|                               | POST      |                                                                    |
|                               | DELETE    |                                                                    |
| /research-subscriptions       | GET       | Creates and manages notifications for research publications.       |
|                               | POST      |                                                                    |
|                               | DELETE    |                                                                    |

## Responses

A HTTP response status code is returned for the indicated HTTP request. Descriptions for each response code is listed below.

| Code | Response Description                                                                                                                                                                                                                                                               |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 200  | <b>Success</b><br>The request was successfully received, understood, and accepted. Matching bucket returned.                                                                                                                                                                       |
| 400  | <b>Bad Request</b><br>Can be the result of malformed request syntax, invalid request message parameters, or deceptive request routing, etc.                                                                                                                                        |
| 403  | <b>Forbidden</b><br>The request is formed correctly, but you do not have the necessary permissions for the resource.                                                                                                                                                               |
| 410  | <b>Gone</b><br>The requested resource has been permanently removed. It is similar to a 404 (Not found) code but is sometimes used in the place of a 404 for resources that used to exist but no longer do.                                                                         |
| 500  | <b>Internal Server Error</b><br>A generic error message indicating that an unexpected condition was encountered on the server side and no more specific message is suitable.<br>Returned most often when a web framework executes a request handler code that raises an exception. |

## Appendix A - Glossary of Terms & Abbreviations

| Abbreviation | Meaning                                                   |
|--------------|-----------------------------------------------------------|
| API          | Application Programming Interface                         |
| AWS          | Amazon Web Service                                        |
| CUSIP        | Committee on Uniform Securities Identification Procedures |
| ESG          | Environmental Social Governance                           |
| GUI          | Graphical User Interface                                  |
| Href         | Hypertext Reference                                       |
| HTTP         | Hypertext Transfer Protocol                               |
| ISIN         | International Security Identification Number              |
| JSON         | JavaScript Object Notation                                |
| LEI          | Legal Entity Identifier                                   |
| PermID       | Permanent ID. For more information see PermID.org         |
| PNAC         | Primary News Access Code                                  |
| RDP          | Refinitiv Data Platform                                   |
| RIC          | Reuters Instrument Codes                                  |
| RIM          | Refinitiv Information Model                               |
| REST         | Representational State Transfer                           |
| SEDOL        | Stock Exchange Daily Official List                        |
| SQS          | Amazon Simple Queue Service                               |
| STS          | Security Token Service                                    |
| UI           | User Interface                                            |
| URI          | Uniform Resource Identifier                               |
| URL          | Uniform Resource Locator                                  |
| UOF          | Universal Order Form                                      |
| ESG          | Environmental Social Governance                           |
| GUI          | Graphical User Interface                                  |

## Appendix B - Reuters Editorial Story Evolution

This section provides a description and examples of how common Stories are created by Reuters Editorial:

- simple Story;
- subsequent Updates;
- resolving errors in Stories

**Note:** This section describes the most common processes. However, due to the time-sensitive and manual nature of journalistic work, exceptions to these rules may occur.

**Note:** Diagrams contain Latin words in order to show the location and format of text.

### Simple Story

When a newsworthy event occurs, the first part of a Story may be an Alert, a short sentence in upper-case that contains the facts and essential detail. Often several Alerts are filed in quick succession.

A Newsbreak is generally created 5-20 minutes after any Alerts. Newsbreaks comprise a Headline (often different from the Alert) and perhaps two to four paragraphs of body text putting the facts into context and making them meaningful.

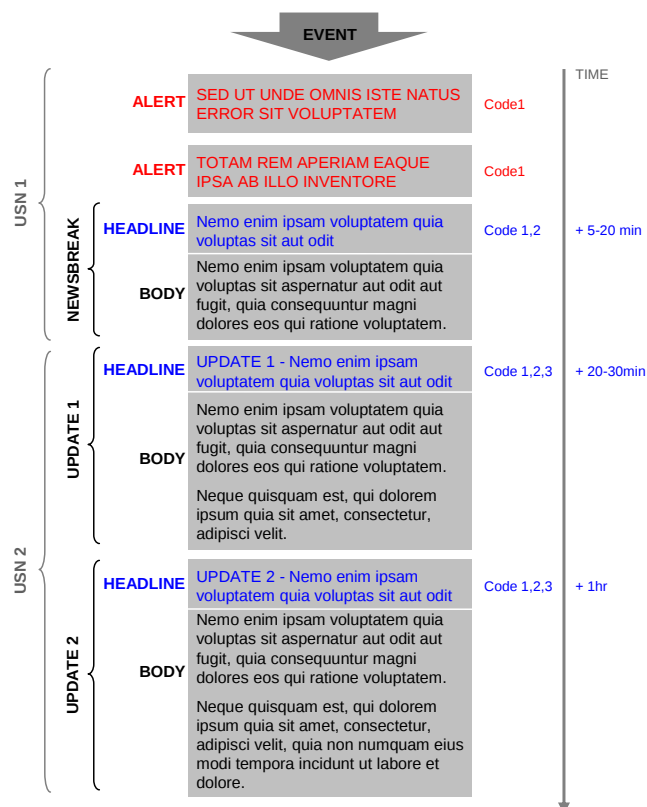
An Update may be filed 20-30 minutes after a Newsbreak. Updates comprise a Headline (sometimes different from the Headline in the original Newsbreak) and 6-20 paragraphs of body text with further information about the event.

The Update may be refreshed as the Story develops. Each subsequent Update replaces the previous Update, but the original Alert(s) and Newsbreak remain.

### Additional Takes

The Story Body together with its associated Headline is called a take. Stories are usually filed in a single take. Occasionally, however, further takes are necessary to add any of the following:

- additional text
- additional codes (e.g., topic codes)
- both additional text and codes
- a cross-reference table



---

**Note:** Although the examples in this section refer to “first take” and “second take”, the process applies equally to further takes (e.g., “fourth take” and “fifth take”).

---

#### Adding text

In cases where text is being added to the Story (but not codes), the second take is filed with the same Headline as the first take, but with the additional text appended in the body.

#### Adding codes

When a second take is used to add codes (but not additional text), it is filed with the same Headline as the first take, with the codes added.

---

**Note:** Product codes do not follow this rule and may not be changed once the first take is sent out.

---

#### Adding text and codes

A second take can also be used to add both additional codes and text.

The second take is filed with the same Headline as the first take, with the codes added and with the additional text appended in the body.

---

**Note:** Note: Product codes do not follow this rule and may not be changed once the first take is sent out.

---

#### Adding a cross-reference table

Cross-reference tables at the end of Stories are used by Thomson Reuters desktop product subscribers to click-through to other news and data.

Journalists sometimes replace the automatically generated cross-references with a customized version.

In this case, a second take is created with the new cross-reference table in the body.

|        |          |                                                                                                                                                  |
|--------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| TAKE 1 | HEADLINE | Nemo enim ipsam voluptatem quia voluptas sit aut odit                                                                                            |
|        | BODY     | Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem.      |
| TAKE 2 | HEADLINE | Nemo enim ipsam voluptatem quia voluptas sit aut odit                                                                                            |
|        | BODY     | Reuters Terminal users can see related statistics and news by double clicking on:<br>Korea equities <KR/EQUITY><br>Gainers/losers <KR/STATS1>... |

## Normal Updates

If Stories need to be revised with additional information (e.g., fresh developments, reaction, added context or interpretation) an Update is issued. The Update may be refreshed as the Story develops. These subsequent Updates are filed either by replacing the previous Update or by appending the latest information.

### Replacing previous

Most subsequent Updates replace the previous Update in the series, and the update number is noted in the Headline tag, e.g., **UPDATE 1**, **UPDATE 2**.

### Appending latest

Sometimes, the text of the subsequent Update is added to the previous Update in the series (similar to the process for an additional take), with a "=2", "=3", etc. added to the end of the Headline to indicate the version.

|        |          |                                                                                                                                                        |
|--------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| TAKE 1 | HEADLINE | <del>UPDATE1-Nemo enim ipsam voluptatem quia voluptas sit aut odit</del>                                                                               |
|        | BODY     | <del>Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem.</del> |
| TAKE 2 | HEADLINE | UPDATE2-Nemo enim ipsam voluptatem quia voluptas sit aut odit                                                                                          |
|        | BODY     | Neque quisquam est, qui dolorem ipsum quia sit amet, consectetur, adipisci velit, quia non numquam eius modi tempora incidunt ut labore.               |
| TAKE 1 | HEADLINE | UPDATE-Nemo enim ipsam voluptatem quia voluptas aut odit                                                                                               |
|        | BODY     | Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem.            |
| TAKE 2 | HEADLINE | UPDATE-Nemo enim ipsam voluptatem quia voluptas aut odit =2                                                                                            |
|        | BODY     | Neque quisquam est, qui dolorem ipsum quia sit amet, consectetur, adipisci velit, quia non numquam eius modi tempora incidunt ut labore.               |

## Updates to Errors

At Thomson Reuters we do our utmost to avoid errors, but any faults are rectified promptly and clearly. We do not disguise or bury mistakes in subsequent Updates or Stories.

One of three processes is followed, depending on the severity of the error:

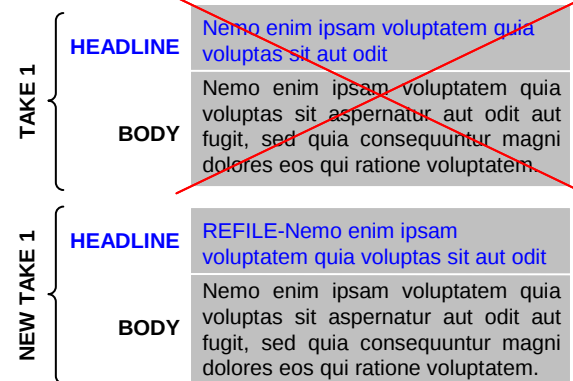
- **Refile** – used to correct minor factual errors in Stories which have no bearing on any investment decision or understanding of the news (e.g., common words misspelled).
- **Corrected/Correction** – terms used to distinguish the two types of corrective action used whenever a substantive, factual error appears in a Story which alters the meaning or significance of the Story or passage (e.g., "million" rather than "billion").
- **Withdrawal** – used for Stories that are so fundamentally flawed that correcting is impossible.

### Refiles

If there is a minor error in a Story it is refiled in one of two ways: replacement or overwrite.

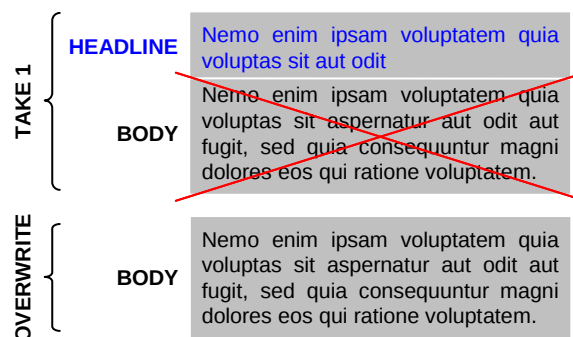
### Replacement

In most refiles, the Headline and body are both republished, with the headline tag **REFILE** (since 2006). Any Alerts in the Story have to be repeated.



### Overwrite

If there are Alerts in the Story that are correct and there is a compelling reason to retain them on screens while refiling the Newsbreak, the Headline of the Story is left unchanged, but the body of the Story is overwritten with the amended version.



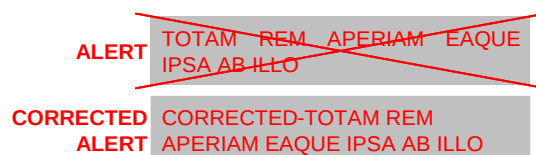
## Corrected and Correction

### Correcting Alerts

If there is a substantial error in an Alert, it is corrected.

When an Alert is corrected, the journalist files it with a new PNAC. **CORRECTED-** is inserted at the beginning of the text, and an explanation of what was changed is included in brackets,

Example: **CORRECTED-TRANSMENIAN INDUSTRIES NET PROFIT 5.4 MLN EUROS (NOT 4.5 MLN)**



### Variations

- If the explanation would make the Alert too long, the details are provided at the top of the Newsbreak
- If there is a series of Alerts on the same PNAC and one or more Alerts are corrected, the other Alerts (where no correction is required) are repeated on the new PNAC along with the corrected Alert (**RPT-** is inserted at the beginning of the text for each of the repeated Alerts)
- For an official correction to an Alert, **CORRECTED-(OFFICIAL)-** is added to the beginning

## Correcting Newsbreaks

Correcting is carried out if there is a substantial error in a Story – either by a correction or by being corrected.

For both, the amended Story is usually filed with the same PNAC, a headline tag **CORRECTED** and an explanation of what has been changed in the advisory line.

### Corrected

If both the Body Text and Headline need to be replaced, a corrected message is issued for a Story.

The amended Story is usually filed with the same PNAC but the Story date/time will be different.

|           |          |                                                                                                                                                        |
|-----------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| TAKE 1    | HEADLINE | <del>Nemo enim ipsam voluptatem quia voluptas sit aut odit</del>                                                                                       |
|           | BODY     | <del>Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem.</del> |
| CORRECTED | HEADLINE | CORRECTED-Nemo enim ipsam voluptatem quia voluptas sit aut odit                                                                                        |
|           | BODY     | Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem.            |

### Correction

If the Headline needs to be replaced but new body text appended, a correction is issued for a Story.

The amended Story is usually filed with the same PNAC and Story date/time.

|            |          |                                                                                                                                                                                                                                      |
|------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TAKE 1     | HEADLINE | <del>Nemo enim ipsam voluptatem quia voluptas sit aut odit</del>                                                                                                                                                                     |
|            | BODY     | Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem.                                                                                          |
| CORRECTION | HEADLINE | CORRECTED-Nemo enim ipsam voluptatem quia voluptas sit aut odit                                                                                                                                                                      |
|            | BODY     | Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem.<br><br>Neque quisquam est, qui dolorem ipsum quia sit amet, consectetur, adipisci velit. |

## Variations

Alerts and Newsbreaks are filed with the same PNAC, so Newsbreaks may be affected if an Alert is corrected:

- If the Newsbreak has not yet been filed, it is filed with the same PNAC as the corrected Alert
- If the Newsbreak has been filed and is correct, it is resent as a repeat to link with the corrected Alert

- If the Newsbreak has been filed and is incorrect, the corrected version is sent out with the new PNAC of the corrected Alert, with **CORRECTED** as the headline tag
- An official correction to a Newsbreak takes a headline tag of **CORRECTED-(OFFICIAL)**

---

**Note:** Any Update after a corrected Story carries a new PNAC and therefore does not replace the amended Story.

---

## Withdrawals

If a Story is fundamentally flawed, a delete message is sent for the original Story which removes it from the screens. The delete message, as well as the original Story, is not available from the archive.