Transparency and Consent String with Global Vendor & CMP List Formats

IAB Europe Transparency & Consent Framework

Final v.2.0 | August 2019, Updated December 2019

Table of Contents

- Version History:
- Introduction
  - Audience
  - Relevant Documents
  - About the Transparency & Consent Framework
  - License
  - About IAB Tech Lab
  - About IAB Europe
- About the Transparency & Consent String (TC String)
  - Definitions
  - What purpose does a TC String serve?
  - What information is stored in a TC String?
  - Who should create a TC string?
  - When should a TC string be created?
  - What are the different scopes for a TC String?
  - What are publisher restrictions?
  - How does the CMP handle a globally-scoped TC string?
  - How does a URL-based service process the TC string when it can't execute JavaScript?
    - Full TC String passing
    - CMP Redirect for TC String
  - What if consent is governed differently in a country?
- Creating a TC String
  - How should a Transparency & Consent String be stored?
  - What are the Purposes and Features being supported?
  - How should a global TC string be formatted for storage?
  - TC String Format
    - The Core String
    - Signaling OOB in the TC String

        - Disclosed Vendors (OOB)
        - Allowed Vendors (OOB)
        - Publisher Purposes Transparency and Consent
- The Global Vendor List
    - I'm a vendor, how do I get added to the Global Vendor List?
    - What is contained in the Global Vendor List?
    - Where can I access the Global Vendor List?
    - TCF version 1 of the Global Vendor List (deprecated)
    - Translations for Purposes, Special Purposes, Features, and Special Features
    - How often is the Global Vendor List updated?
    - CMPs using the GVL
    - Vendors using the GVL
    - Accessing And Caching the Global Vendor List
        - CMPs accessing and caching the GVL
        - Vendors accessing and caching the GVL
        - Using a compressed version of the Global Vendor List
        - Global Vendor List and TCF Policy Updates
    - Example Global Vendor List JSON Object
- Global CMP List
    - What is contained in the Global CMP List?
    - Where can I access the Global CMP List?
    - How often is the Global CMP List updated?
    - Caching the Global CMP List
    - Server-side caching of the GCL
    - Using a compressed version of the Global CMP List
    - Example Global CMP List JSON Object

Version History:

Date	Version	Comments
May 2020	2.0	Updated to clarify questions on `RestrictionType` cases
December 2019	2.0	Updated with global cookie support notes, Updated macros to be upper case
August 2019	2.0	Version 2.0 released to the public
April 2019	2.0	Released for public comment
April 2018	1.1	First version released to the public

Introduction

This document is one of the IAB Europe Transparency and Consent Framework Specifications. It defines the technical implementation of the structure and encoding for a Transparency and Consent String (TC String), and the format for a Global Vendor List (GVL) maintained by IAB Europe. The TC String is a technical component of the IAB Europe Transparency & Consent Framework (TCF).

The General Data Protection Regulation (GDPR) requires a high level of accountability for how personal data is processed. While important to all parties in the digital advertising ecosystem, implementation of the GDPR came with heavy technical challenges.

The GDPR requires, amongst others, a legal basis for such processing. The two most relevant legal bases are the consent of the user to the processing of their personal data, and the legitimate interests of the controller or a third party to the processing of a user's personal data, provided that the interests and fundamental rights of the user are not overriding. Both legal bases require the provision of disclosures to ensure transparency, and the opportunity for user choice either through the user's consent to the processing of their personal data before the processing starts if the legal basis is consent, or through the user's objection to the processing of their personal data after the processing starts if the legal basis is a legitimate interest. Under the GDPR, controllers are required to create and maintain records of compliance, including, but not limited to user consent records. This warrants clear standards for a common technical solution for all affected parties and policies to govern how that solution is used.

IAB Europe established the TCF to support compliance with the GDPR in the context of digital advertising. This framework is built on four components: a Global Vendor List (GVL), a Transparency and Consent String (TC String), an API for Consent Management Providers (CMPs) to create and process the TC String, and the Policies that govern how the TCF is used.

Prescribed use of the TCF may support compliance with the GDPR, but the real benefit to the digital advertising ecosystem is a safer Internet for consumers, and more reliable data for brands and publishers. As adoption of the TCF increases, compliance becomes more scalable and data becomes more meaningful.

To participate in the use of the TCF, vendors must make a public attestation of compliance with the Policies for using it. To have transparency and consent established and signaled status for your online services stored in a global database, apply to be added to the GVL. To play a role in creating a TC String for signaling status on transparency and user consent, sign up with IAB Europe to become a CMP. CMPs must follow technical standards provided in this document for creating TC Strings in compliance with TCF Policies. They must also follow technical standards guidance for using the CMP API specified in this document to receive and process information provided in a TC String.

Audience

Engineers for a registered CMP can use this document to design or update a solution for generating a TC String. In particular, first parties (content publishers, advertisers, and other suppliers of online services) and third-party (vendors for data-driven services) organisations should be familiar with the purpose and scope of a TC String as well as what information it provides, and support its implementation.

Relevant Documents

IAB Europe Transparency & Consent Framework Policies

Consent Manager Provider JS API

About the Transparency & Consent Framework

IAB Europe Transparency & Consent Framework (TCF) has a simple objective to help all parties in the digital advertising chain ensure that they comply with the EU's General Data Protection Regulation and ePrivacy Directive when processing personal data or accessing and/or storing information on a user's device, such as cookies, advertising identifiers, device identifiers and other tracking technologies. IAB Tech Lab stewards the development of these technical specifications.

Resources including policy FAQ, Global Vendor List, and CMP List can be found at iabeurope.eu/tcf.

License

Disclaimer

THE STANDARDS, THE SPECIFICATIONS, THE MEASUREMENT GUIDELINES, AND ANY OTHER MATERIALS OR SERVICES PROVIDED TO OR USED BY YOU HEREUNDER (THE "PRODUCTS AND SERVICES") ARE PROVIDED "AS IS" AND "AS AVAILABLE," AND IAB TECHNOLOGY LABORATORY, INC. ("TECH LAB") MAKES NO WARRANTY WITH RESPECT TO THE SAME AND HEREBY DISCLAIMS ANY AND ALL EXPRESS, IMPLIED, OR STATUTORY WARRANTIES, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AVAILABILITY, ERROR-FREE OR UNINTERRUPTED OPERATION, AND ANY WARRANTIES ARISING FROM A COURSE OF DEALING, COURSE OF PERFORMANCE, OR USAGE OF TRADE. TO THE EXTENT THAT TECH LAB MAY NOT AS A MATTER OF APPLICABLE LAW DISCLAIM ANY IMPLIED WARRANTY, THE SCOPE AND DURATION OF SUCH WARRANTY WILL BE THE MINIMUM PERMITTED UNDER SUCH LAW. THE PRODUCTS AND SERVICES DO NOT CONSTITUTE BUSINESS OR LEGAL ADVICE. TECH LAB DOES NOT WARRANT THAT THE PRODUCTS AND SERVICES PROVIDED TO OR USED BY YOU HEREUNDER SHALL CAUSE YOU AND/OR YOUR PRODUCTS OR SERVICES TO BE IN COMPLIANCE WITH ANY APPLICABLE LAWS, REGULATIONS, OR SELF-REGULATORY FRAMEWORKS, AND YOU ARE SOLELY RESPONSIBLE FOR COMPLIANCE WITH THE SAME.

About IAB Tech Lab

The IAB Technology Laboratory (Tech Lab) is a non-profit consortium that engages a member community globally to develop foundational technology and standards that enable growth and trust in the digital media ecosystem.. Comprised of digital publishers, ad technology firms, agencies, marketers, and other member companies, IAB Tech Lab focuses on improving the digital advertising supply chain, measurement, and consumer experiences, while promoting responsible use of data. Its work includes the OpenRTB real-time bidding protocol, ads.txt anti-fraud specification, Open Measurement SDK for viewability and verification, VAST video specification, and DigiTrust identity service. Board members include ExtremeReach, Facebook, Google, GroupM, Hearst Digital Media, Index Exchange, Integral Ad Science, LinkedIn, LiveRamp, MediaMath, Microsoft, Oracle Data Cloud, Pandora, PubMatic, Quantcast, Rakuten Marketing, Telaria, The Trade Desk, Verizon Media Group, Xandr, and Yahoo! Japan. Established in 2014, the IAB Tech Lab is headquartered in New York City with staff in San Francisco, Seattle, and London. Learn more at https://www.iabtechlab.com.

About IAB Europe

IAB Europe is the European-level association for the digital marketing and advertising ecosystem. Through its membership of National IABs and media, technology and marketing companies, its mission is to lead political representation and promote industry collaboration to deliver frameworks, standards and industry programmes that enable business to thrive in the European market.

Learn more about IAB Europe here: https://www.iabeurope.eu/

About the Transparency & Consent String (TC String)

In the TCF, a TC String is used to encapsulate relevant details about how transparency and consent was established and encoded as it applies for each of the Purposes, Special Purposes, Features, and Special Features defined by the Policies and for participating Vendors. This document specifies how that string must be formatted, who should use it, and how it must be used.

Definitions

Regarding specific definitions as they relate to TCF Policies and the technology described in this document, please refer to IAB Europe Transparency and Consent Framework Policies located at the following link:

https://iabeurope.eu/iab-europe-transparency-consent-framework-policies/

What purpose does a TC String serve?

A TC String's primary purpose is to encapsulate and encode all the information disclosed to a user and the expression of their preferences for their personal data processing under the GDPR. Using a Consent Management Platform (CMP), the information is captured into an encoded and compact HTTP-transferable string. This string enables communication of transparency and consent information to entities, or "vendors", that process a user's personal data. Vendors decode a TC String to determine whether they have the necessary legal bases to process a user's personal data for their purposes. The concise string data format enables a CMP to persist and retrieve a user's preferences any time they're needed as well as transfer that information to any vendors who need it.

What information is stored in a TC String?

A TC String contains the following information:

1. **General metadata:** standard markers that indicate details about a TC String such as its encoding version, when it was last updated, and when it was initially created as well as details about the conditions of the transparency and consent values it contains such as the Global Vendor List version used, the CMP used, etc.
2. **User consent:** a user's expression of consent given for processing their personal data. A user's

consent is expressed on two levels: per Purpose and per Vendor.

3. **Legitimate interest:** the record of a CMP having established legitimate interest transparency for a vendor and/or purpose and whether the user exercised their "Right to Object" to it. This includes signals for Purposes in general and Purposes declared specifically for a given Vendor.
4. **Publisher restrictions:** the restrictions of a vendor's data processing by a publisher within the context of the users trafficking their digital property.
5. **Publisher transparency and consent:** a segment of a TC String that publishers may use to establish transparency with and receive consent from users for their own legal bases to process personal data or to share with vendors if they so choose.
6. **Out-of-band (OOB) legal bases:** two segments expressing that a Vendor is using legal bases outside of the TCF to process personal data. The first segment is a list of Vendors disclosed to the user and the second is a list of Vendors that the publisher allows to use out-of-band legal bases.
7. **Specific jurisdiction disclosures:** the country in which the publisher's business entity is established or the legislative country of reference and a record of whether Purpose 1, "[to] store and/or access information on a device," was disclosed to the user since some jurisdictions handle this Purpose differently.

Who should create a TC string?

A Transparency & Consent String may only be created by an IAB Europe TCF registered CMP using its assigned CMP ID number in accordance with the Policies. Vendors or any other third-party service providers must neither create nor alter TC Strings. These and other requirements are articulated in the Policies to which all parties including CMPs, Publishers, and Vendors, are bound.

When should a TC string be created?

A TC String that contains positive consent signals must not be created before clear affirmative action is taken by a user that unambiguously signifies that user's consent. However, a TC String may be created with only legitimate interest establishment signals providing that legitimate interest transparency has been established in accordance with the Policies.

What are the different scopes for a TC String?

There are two main contexts in which a TC String can be created:

- **Global** - A TC String in this context is saved globally and is shared by CMPs running on sites across the web; When stored globally, they must NOT contain Publisher restrictions or a ***Publisher TC*** segment but they may contain a ***DisclosedVendors*** segment.
- **Service-specific** - A TC String in this context is only used by the site(s) or app(s) on which it is running. One is created for every user on a given site/app or group of sites/apps. They may contain Publisher restrictions, a ***Publisher TC*** segment and an ***AllowedVendors*** segment.

CMPs must be set up to operate in either a service-specific or global configuration. If a Publisher-operated CMP declares that the personal data processing purpose is, for example, on this site and on other sites or apps where third-party companies also operate, then the scope is global and that TC String is used and stored in a global context.

If the disclosures do not describe a global scope, or explicitly state service-specific processing, then the TC String is used and stored explicitly as a service-specific string. Also, if the CMP discloses transparency and consent in a global context but the user's browser does not permit third-party cookies, then the CMP's only recourse is to retain the user's preference using a local storage mechanism (eg. first-party cookie or window.localStorage). Since the transparency and consent obtained from the user is restricted

to that site or service, the TC String must then have the service-specific bit IsServiceSpecific set.

What are publisher restrictions?

Version 2.0 of the Framework introduced the ability for publishers to signal restrictions on how vendors may process personal data:

- **Purposes.** Restrict the purposes for which personal data is processed by a vendor.
- **Legal basis.** Specify the legal basis upon which a publisher requires a vendor to operate where a vendor has signaled flexibility on legal basis in the GVL.

Publisher restrictions are custom requirements specified by a publisher and must only be saved to a service-specific TC String.

How does the CMP handle a globally-scoped TC string?

When configured to use globally-scoped TC Strings CMPs must not overwrite any of the consent or legitimate interest signals found in an existing TC String. Therefore CMPs must do the following:

- Decode the TC String from the global scope to load and preserve all existing signals
- Set the signals for the vendors specified in the CMP user interface. If a subset of vendors is shown in the CMP user interface, the CMP must only set signals for those vendors.
- If a CMP is unable to resolve an ambigious negative vendor signal – unable to differentiate between a "no" and a "never disclosed" – a CMP shall disambiguate the signal with the corresponding value in the ***DisclosedVendors*** segment since that segment signals which vendors were disclosed to the user.
- Once the user has made their selections the CMP shall save the resulting TC String back to the global context, overwriting the old one.

How does a URL-based service process the TC string when it can't execute JavaScript?

When a creative is rendered, it may contain a number of pixels under `<img>` tags. For example, `<img src="http://vendor-a.com/key1=val1&key2=val2">` which fires an HTTP GET request from the browser to Vendor A's domain.

Since the pixel is in an `<img>` tag without the ability to execute JavaScript, the CMP API cannot be used to obtain the TC String. All parties in the ad supply chain who transact using URLs must add a macro in their URLs where the TC String is inserted. Any caller with access to the applicable TC String must insert it within a URL containing the macro `${GDPR_CONSENT_XXXXX}` where `XXXXX` is the numeric Vendor ID of the vendor receiving the TC string.

For example, for Vendor A with ID 123 to receive a TC String, an image URL must include a key-value pair with the URL parameter and macro `gdpr_consent=${GDPR_CONSENT_123}`.

The resulting URL is:

`http://vendor-a.com/key1=val1&key2=val2&gdpr_consent=${GDPR_CONSENT_123}`

If the TC String is:
`COvFyGBOvFyGBAbAAAENAPCAAOAAAAAAAAAAAEEUACCKAAA.IFoEUQQgAIQwgIwQABAEAAAAOIAACAIAAAAQAIA`
`gEAACEAAAAAgAQBAAAAAAAGBAAgAAAAAAAFAAECAAAgAAQARAEQAAAAAJAAIAAgAAAYQEAAAQmAgBC3ZAYzUw`

Then the caller replaces the macro in the URL with the actual TC String so that the originally placed pixel containing the macro is modified as follows when making the call to the specified server.

```
http://vendor-a.com/key1=val1&key2=val2&gdpr_consent=COvFyGBOvFyGBAbAAAENAPCAAOAAAAAAAAAAAEEUACCKAAA.IFoEUQQgAIQwgIwQABAEAAAAOIAACAIAAAAQAIAgEAACEAAAAAgAQBAAAAAAAGBAAgAAAAAAAFAAECAAAgAAQARAEQAAAAAJAAIAAgAAAYQEAAAQmAgBC3ZAYzUw
```

TC Strings must always be propagated as is, and not modified. Additional URLs in the supply chain are addressed the same way with remaining vendors.

The available URL parameters and macros to relay information down the supply chain are listed in the following section.

Full TC String passing

Services that are called using a URL from the user's browser, like cookie staplers, user id associators, and tracking pixels (the 'callee') are passed as macros within the URL and formatted as:

```
&url_parameter=${macro_name}
```

The supported URL parameters and the corresponding macros are defined below:

URL parameter	Corresponding Macro	Representation in URL
`gdpr`	`GDPR`	`&gdpr=${GDPR}`
`gdpr_consent`	`GDPR_CONSENT_XXXXX` (`XXXXX` is numeric Vendor ID - the ID of the vendor on the GVL who is expecting this URL call)	`&gdpr_consent=${GDPR_CONSENT_XXXXX}` E.g. `&gdpr_consent=${GDPR_CONSENT_123}` for Vendor ID 123.
`gdpr_pd`	`GDPR_PD`	`&gdpr_pd=${GDPR_PD}`

The service making the call must replace the macros with appropriate values described in the table below. For macro `${GDPR_CONSENT_XXXXX}`, the service making the call must also check that the macro name contains a valid Vendor ID before replacing the macro. The creator of the URL should ensure these parameters are added only once, and are passed to services which are expecting them and can handle them properly.

Macro	possible values	purpose
`${GDPR}`	`0` / `1`	`0` GDPR does not apply; `1` GDPR applies. If not present, callee should do geoIP lookup, and GDPR applies for EU IP addresses
`${GDPR_CONSENT_XXXXX}`	URL-safe base64-encoded Transparency & Consent string. Only meaningful if `gdpr=1`	Encodes the TC string, as obtained from the CMP JS API or OpenRTB.
`${GDPR_PD}`	`0` / `1` (optional, default: `1`)	for generic URL parameters, `gdpr_pd=0` indicates none of them contain personal data (from the perspective of the callee). For "defined" URL parameters, their definition should define whether they include personal data.

Note: other personal data, like IP addresses or callee cookies, may be passed as part of the request, and the `gdpr` and `gdpr_consent_xxxxx` is used by the callee to determine whether an identifier cookie or other personal data can be set and/or used.

CMP Redirect for TC String

CMPs can implement a consent redirector and host it at `https://[cmpname].mgr.consensu.org/consent?redirect=url`. This redirector can read the (web-wide global) consent cookie which the browser sends with a 302 HTTP redirect URL using the parameters described in the previous section.

What if consent is governed differently in a country?

Policies require consent for Purpose 1 to store and/or access information on a device "where such consent is necessary" leaving the responsibility to publishers and vendors to determine if consent in those jurisdictions is required or not.

If a publisher is operating a CMP within a jurisdiction that does not require consent to store and/or access information on a device and, therefore, does not ask for consent on behalf of a vendor, the CMP will write the corresponding bit in the ***PurposesConsent*** field to `0`. Even though it is valid within that jurisdiction to use Legitimate Interest for Purpose 1, a vendor would interpret that `0` as a "no consent" signal and have no way of knowing that consent was not required in the jurisdiction in which the publisher operates. This lack of transparency would, ultimately, cause losses in ad revenue for that publisher.

To accommodate cases where Purpose 1 is governed differently for consent depending on the jurisdiction, a TC String is transparent about the publisher's operating governance and whether or not Purpose 1 was disclosed to a user. The vendor can then use these details to make a determination about whether they have sufficient legal bases for personal data processing in that given context. To support this, there are two fields in a TC String: ***PublisherCC***, which represents the publisher's country code and a flag for whether any disclosure has been offered on Purpose 1 named ***PurposeOneTreatment***. Details for each field are listed among the fields used in the TC String.

Creating a TC String

The following details provide information on creating, storing, and managing a TC String.

How should a Transparency & Consent String be stored?

In version 1 of the TCF Specifications the consent string was specified to be stored as either a 1st party cookie for service-specific consent or a 3rd party cookie for global consent. In version 2 of the TCF Specifications, the storage mechanism used for service-specific TC Strings is up to a CMP, including any non-cookie storage mechanism. However, global TC Strings must still be stored as cookies under the `consensu.org` domain.

It is important to note that with the creation of the version 2 TCF Specifications globally-scoped and service-specific scoped TC Strings have different encoding and decoding requirements. Some segments are not allowed in a global scope and some are not allowed in a service-specific scope. This document attempts to call out those differing requirements explicitly where applicable.

The following table summarises where data is stored:

Scope	Storage	Purpose
Global	3rd-party .consensu.org cookie. CMPs may also "backup" a TC String encoded for the global scope via a different storage mechanism if 3rd-party cookies are being blocked or erased by a browser.	Web-wide vendor transparency & consent
Service-specific	Storage mechanism chosen by CMP. Must not be stored as the version 1.1 local 'euconsent' cookie.	Service-specific vendor transparency & consent (if configured, overrides global vendor transparency & consent)

Managing conflicting string versions

Before 30 September 2020, after which v1.x strings will be considered invalid, if a CMP encounters a situation where both a v1.x string and a v2.0 string are erroneously present simultaneously, the CMP should remove the v1.x string to ensure that there is only one source of truth for consumers of the string.

Note: TCF version 2 introduces "Publisher Restrictions", which, if exhausted by a publisher, could result in TC strings that are larger than the size limit for cookies. While this possibility is remote, it should be guarded against – a CMP should work with a publisher to help them accomplish their goals. Publisher Restrictions are only allowed in TC Strings, therefore within a service-specific context so CMPs may need to take this into consideration when deciding on the storage mechanism for those TC Strings.

What are the Purposes and Features being supported?

The IAB Europe Transparency & Consent Framework Policies defines Purposes, Special Purposes, Features, Special Features, and Stacks (groupings of Purposes and/or Special Features). You can reference the details of these purposes and features in the document found at the following URL:

https://iabeurope.eu/iab-europe-transparency-consent-framework-policies/

How should a global TC string be formatted for storage?

The global TC string is stored in a shared space and is formatted as described in the following table:

Cookie Directive	Value(s)	Notes
Name	`euconsent-v2`	To avoid conflicts with TC String cookie storage, beginning with version 2.0 of the TCF the global and service-specific cookie name shall include the TC string version as a hyphenated postfix, for example `euconsent-v2`.
Host	`.consensu.org`	The DNS resolution for the name `[cmp-name].mgr.consensu.org` will be delegated by the Managing Organisation (IAB Europe) to each CMP. CMPs will host their code, APIs, and CDN under this domain or subdomains.
Path	`/`	
Max-Age	`33696000`	This represents thirteen 30-day months.
Value	Encoded TC String	

Global Cookie Storage Update (December 2019)

- All requests that read from or write to the global cookie in the consensu.org domain must be secured by HTTPS
- Additionally, browser cookie policies may require the support of certain attributes (e.g. sameSite, Secure)

TC String Format

There are 4 distinct TC String segments that are joined together on a “dot” character. They are:

- The core vendor transparency and consent details
- Disclosed vendors for validating OOB signaling
- Allowed vendors for restricting OOB signaling to select vendors, and
- Publisher purposes transparency and consent for their own data uses.

The ***Core String*** is always required and comes first and includes all the details required for communicating basic vendor transparency and consent. The remaining optional and arbitrarily ordered segments represent support for out-of-band (OOB) signaling and publisher purposes transparency and consent (publisher TC). A TC String with all four segments is possible in certain conditions.

For example, a globally-scoped TC String with all four segments present would be surfaced through CMP API – not stored – and look like:

[***Core String***].[***Disclosed Vendors***].[***AllowedVendors***].[***Publisher TC***]

```
COw4XqLOw4XqLAAAAAENAXCAAAAAAAAAAAAAAAAAAAA.IFukWSQgAIQwgI0QEByFAAAAeIAACAIgSAAQAIA
gEQACEABAAAgAQFAEAIAAAGBAAgAAAAQAIFAAMCQAAgAAQiRAEQAAAAANAAIAAggAIYQFAAARmggBC3ZCYzU
2yIA.QFukWSQgAIQwgI0QEByFAAAAeIAACAIgSAAQAIAgEQACEABAAAgAQFAEAIAAAGBAAgAAAAQAIFAAMCQ
AAgAAQiRAEQAAAAANAAIAAggAIYQFAAARmggBC3ZCYzU2yIA.YAAAAAAAAAAAAAAAAAA
```

A service-specific TC String must contain a Core TC String and may optionally contain a ***Publisher TC*** segment, but must not contain the OOB-related segments because those segments are not allowed in service-specific contexts:

[***Core String***].[***Publisher TC***]

```
CLcVDxRMWfGmWAVAHCENAXCkAKDAADnAABRgA5mdfCKZuYJez-
NQm0TBMYA4oCAAGQYIAAAAAAEAIAEgAA.argAC0gAAAAAAAAAAAA
```

The Core String

The following fields are stored in big-endian format. Bit numberings are left-to-right.

Field Name	Bits	Value(s)	Notes
Version	6 bits	Version number of the encoding format	the value is 2 for this format.
Created	36 bits	Epoch deciseconds when this TC String was first created (should not be changed unless a new TCString is created from scratch)	A decisecond is 1/10th of a second. To create a decisecond timestamp in JavaScript: `Math.round((new Date()).getTime()/100)`
LastUpdated	36 bits	Epoch deciseconds when TC String was last updated (Must be updated any time a value is changed)	
CmpId	12 bits	Consent Management Platform ID that last updated the TC String	A unique ID will be assigned to each Consent Management Platform.
CmpVersion	12 bits	Consent Management Platform version of the CMP that last updated this TC String	Each change to a CMP should increment their internally assigned version number as a record of which version the user gave consent and transparency was established.

ConsentScreen	6 bits	CMP Screen number at which consent was given for a user with the CMP that last updated this TC String	The number is a CMP internal designation and is CmpVersion specific. The number is used for identifying on which screen a user gave consent as a record.
ConsentLanguage	12 bits	Two-letter ISO 639-1 language code in which the CMP UI was presented	Each letter is encoded as 6 bits, a=0..z=25.
VendorListVersion	12 bits	Number corresponds to GVL `vendorListVersion`	Version of the GVL used to create this TC String.
TcfPolicyVersion	6 bits	Version of policy used within GVL	From the corresponding field in the GVL that was used for obtaining consent. A new policy version invalidates existing strings and requires CMPs to re-establish transparency and consent from users.
IsServiceSpecific	1 bit	`1` true `0` false	Whether the signals encoded in this TC String were from service-specific storage (`true`) versus 'global' consensu.org shared storage (`false`).
UseNonStandardStacks	1 bit	`1` CMP used non-IAB standard stacks during consent gathering `0` IAB standard stacks were used	Setting this to 1 means that a publisher-run CMP – that is still IAB Europe registered – is using customized Stack descriptions and not the standard stack descriptions defined in the Policies (Appendix A section E). A CMP that services multiple publishers sets this value to `0` .
			The TCF Policies designates certain

SpecialFeatureOptIns	12 bits	One bit for each Special Feature: `1` Opted in `0` Not opted in	Features as “special” which means a CMP must afford the user a means to opt in to their use. These “Special Features” are published and numerically identified in the Global Vendor List separately from normal Features.
PurposesConsent (renamed from PurposesAllowed)	24 bits	One bit for each Purpose: `1` Consent `0` No Consent	The user’s consent value for each Purpose established on the legal basis of consent. The Purposes are numerically identified and published in the Global Vendor List. From left to right, Purpose 1 maps to the `0`th bit, purpose 24 maps to the bit at index 23. Special Purposes are a different ID space and not included in this field.
PurposesLITransparency	24 bits	One bit for each Purpose: `1` legitimate interest established `0` legitimate interest was **NOT** established or it was established but user exercised their “Right to Object” to the Purpose	The Purpose’s transparency requirements are met for each Purpose on the legal basis of legitimate interest and the user has not exercised their “Right to Object” to that Purpose. By default or if the user has exercised their “Right to Object” to a Purpose, the corresponding bit for that Purpose is set to `0`. From left to right, Purpose 1 maps to the 0th bit, purpose 24 maps to the bit at index 23. Special Purposes are a different ID space and not included in this field.
Specific Jurisdiction Disclosures			
			CMPs can use the

PurposeOneTreatment	1 bit	`1` Purpose 1 was NOT disclosed at all. `0` Purpose 1 was disclosed commonly as consent as expected by the Policies.	PublisherCC field to indicate the legal jurisdiction the publisher is under to help vendors determine whether the vendor needs consent for Purpose 1. In a globally-scoped TC string, this field must always have a value of 0. When a CMP encounters a globally-scoped TC String with PurposeOneTreatment=1 then it is considered invalid and the CMP must discard it and re-establish transparency and consent.
PublisherCC	12 bits	ISO 3166-1 alpha-2 code	The country code of the country that determines legislation of reference. Commonly, this corresponds to the country in which the publisher's business entity is established. Each letter is encoded as 6 bits, a=0..z=25.
Vendor Consent Section			
MaxVendorId	16 bits	The maximum Vendor ID that is represented in the following bit field or range encoding.	Because this section can be a variable length, this indicates the last ID of the section so that a decoder will know when it has reached the end.
IsRangeEncoding	1 bit	`1` Range `0` BitField	The encoding scheme used to encode the IDs in the section – Either a BitField Section or Range Section follows. Encoding logic should choose the encoding scheme that results in the smaller output size for a given set.

BitField Section		**Encodes one consent bit per Vendor ID**	
BitField	MaxVendorId bits	One bit for each Vendor: `1` Consent `0` No Consent	The consent value for each Vendor ID from `1` to MaxVendorId where index `0` is Vendor ID `1`. Set the bit corresponding to a given vendor to `1` if the user has consented to this vendor processing their personal data
Range Section			**Encodes range groups of Vendor IDs who have received consent from a user**
NumEntries	12 bits	Number of RangeEntry sections to follow	
RangeEntry (repeated NumEntries times)		A single or range of Vendor ID(s) who have received consent. If a Vendor ID is not within the bounds of the ranges then the vendor is assumed to have "No Consent".	
IsARange	1 bit	`1` Vendor ID range `0` Single Vendor ID	If more than one Vendor ID is included in this RangeEntry then this describes a range of Vendor IDs and this value is 1. If only one Vendor ID is included then the value is 0.
StartOrOnlyVendorId	16 bits	The first ID of an inclusive contiguous ascending-order series of Vendor IDs even if the series is only a cardinality of 1.	This is the first or only Vendor ID with consent in this RangeEntry.
EndVendorId	16 bits	The last ID of the inclusive contiguous ascending-order series of Vendor IDs started with StartOrOnlyVendorId but only if that series has a cardinality greater than 1, otherwise this field is	The end of the series of Vendor IDs – this is omitted if `IsARange=0`.

		omitted.	
Repeated RangeEntry sections to NumEntries...			
Vendor Legitimate Interest Section			
MaxVendorId	16 bits	The maximum Vendor ID that is represented in the following bit field or range encoding.	Because this section can be a variable length, this indicates the last ID of the section so that a decoder will know when it has reached the end.
IsRangeEncoding	1 bit	`1` Range `0` BitField	The encoding scheme used to encode the IDs in the section – Either a BitField Section or Range Section follows. Encoding logic should encode with the encoding scheme that results in the smaller output size for a given set.
BitField Section		**Encodes one legitimate interest bit per Vendor ID**	
BitField	MaxVendorId bits	One bit for each Vendor: `1` Legitimate Interest established `0` Legitimate Interest not established or the user exercised their “Right to Object”	The legitimate interest value for each Vendor ID from `1` to MaxVendorId where index `0` is Vendor ID `1`. Set the bit corresponding to a given vendor to `1` if the CMP has established transparency for that vendor's legitimate interest disclosures for one or more Purposes (not Special Purposes). If a user exercises their “Right To Object” to a vendor’s processing based on a legitimate interest, then that vendor’s bit must be set to `0`.
Range Section		**Encodes range groups of Vendor IDs who have established their legitimate interest disclosures with a user**	

NumEntries	12 bits	Number of RangeEntry sections to follow	
RangeEntry (repeated NumEntries times)		A single or range of Vendor ID(s) who have established transparency for their legitimate interest disclosures with the user. If a Vendor ID is not within the bounds of the ranges then they have not established that transparency.	
IsARange	1 bit	`1` Vendor ID range `0` Single Vendor ID	If more than one Vendor ID is included in this RangeEntry then this describes a range of Vendor IDs and this value is `1`. If only one Vendor ID is included then the value is `0`.
StartOrOnlyVendorId	16 bits	The first ID of an inclusive contiguous ascending-order series of Vendor IDs even if the series is only a cardinality of 1.	This is the first or only Vendor ID with legitimate interest disclosures established in this RangeEntry.
EndVendorId	16 bits	The last ID of the inclusive contiguous ascending-order series of Vendor IDs started with StartOrOnlyVendorId but only if that series has a cardinality greater than `1`, otherwise this field is omitted.	The end of the series of Vendor IDs – this is omitted if `IsARange=0`.
Repeated RangeEntry sections to NumEntries...			
Publisher Restrictions Section	**The content of this section is optional EXCEPT for NumPubRestrictions. Encodes any number of single or range restriction entries**		
NumPubRestrictions	12 bits	Number of restriction records to follow. **Value is required** even if it is `0`	
PubRestrictionEntry (Repeated NumPubRestrictions times)			Each Publisher Restriction Entry is made up of three parts: Purpose ID, Restriction Type and, List

			of Vendor IDs under that Purpose restriction.
PurposeId	6 bits	Purpose ID	The Vendor's declared Purpose ID that the publisher has indicated that they are overriding.
RestrictionType	2 bits	Enum `0` Purpose Flatly Not Allowed by Publisher (regardless of Vendor declarations) `1` Require Consent (if Vendor has declared the Purpose IDs legal basis as Legitimate Interest and flexible) `2` Require Legitimate Interest (if Vendor has declared the Purpose IDs legal basis as Consent and flexible)	Vendors must always respect a `0` (Not Allowed) regardless of whether or not they have not declared that Purpose to be "flexible". Values `1` and `2` are in accordance with a vendors declared flexibility. Eg. if a vendor has Purpose 2 declared as Legitimate Interest but also declares that Purpose as flexible and this field is set to `1`, they must then check for the "consent" signal in the VendorConsents section to make a determination on whether they have the legal basis for processing user personal data under that Purpose. When a vendor's Purpose registration ***is not flexible*** they should interpret this value in the following ways: If this value is `1` and vendor is registered under Legitimate Interest for that Purpose then the vendor *should not process* for that Purpose. If this value is `1` and vendor is registered under Consent for that Purpose then the vendor *can ignore* the signal.

		`3` UNDEFINED (not used)	If this value is `2` and vendor is registered under Consent for that Purpose then the vendor *should not process* for that Purpose. If this value is `2` and vendor is registered under Legitimate Interest for that Purpose then the vendor *can ignore* the signal. If this value is `1` or `2` and the vendor is not registered for the Purpose then the vendor *should not porcess* for that Purpose. **Note:** Purpose 1 is always required to be registered as a consent purpose and can not be flexible per Policies.
NumEntries	12 bits	Number of RangeEntry sections to follow.	
RangeEntry (repeated NumEntries times)		A single or range of Vendor ID(s) who the publisher has designated as restricted under the Purpose ID in this PubRestrictionsEntry.	
IsARange	1 bit	`1` Vendor ID range `0` Single Vendor ID	If more than one Vendor ID is included in this RangeEntry then this describes a range of Vendor IDs and this value is `1`. If only one Vendor ID is included then the value is `0`.
StartOrOnlyVendorId	16 bits	The first ID of an inclusive contiguous ascending-order series of Vendor IDs even if the series is only a cardinality of 1.	This is the first or only Vendor ID with this restriction in this RangeEntry
		The last ID of the inclusive contiguous ascending-order series of Vendor IDs	

EndVendorId	16 bits	started with StartOrOnlyVendorId but only if that series has a cardinality greater than 1, otherwise this field is omitted.	The end of the series of Vendor IDs – this is omitted if `IsARange=0`.
Repeated RangeEntry sections to NumEntries...			
Repeated PubRestrictionsEntry sections to NumPubRestrictions...			

Signaling OOB in the TC String

On occasion, legal bases for processing a user's personal data are achieved outside of the TCF. This would be considered an out-of-band (OOB) legal basis. To signal whether using an OOB legal bases is allowed requires:

- An indication that some CMP has, at some time, disclosed the vendor in a global context to the user in the ***DisclosedVendors*** segment
- The use of a global-context TC String
- The publisher to allow vendors, in general, to use OOB legal bases
- Optionally, a list of specific vendors allowed to use OOB legal bases in the ***AllowedVendors*** segment

The ***DisclosedVendors*** segment of a TC String provides a list of vendors that have been disclosed to a user; it is created and stored in a global context for all CMPs to share across the web. The existence of this segment as a member of a TC String, when signaling, implies that the publisher supports OOB legal bases. Conversely, If a publisher does not support OOB legal bases the segment shall be omitted when signaling. Regardless of publisher support, a CMP shall still update the segment with any new Vendor IDs disclosed and save the updated TC String back to the global context when the CMP user interface completes its interaction with the user.

If a publisher supports OOB legal bases, but only for select vendors, a CMP shall create an ***AllowedVendors*** segment that reflects the vendors the publisher allows to operate under OOB legal bases. When a TC String is requested from the CMP API it shall include both the ***AllowedVendors*** and ***DisclosedVendors*** segments. However, when a TC String is stored, an ***AllowedVendors*** segment must never be saved to the global context as this is a publisher-specific setting and does not apply web-wide. If a CMP encounters a TC String with an ***AllowedVendors*** segment in the global context it must disregard it, not include it in responses from the CMP API, and of course omit it when re-saving.

Note: If a Vendor has been *disclosed* within the ***DisclosedVendors*** segment that means that they have interacted with the Framework and therefore can not use OOB legal bases.

The following three examples demonstrate how to handle an OOB signal in the TC String.

Example 1: A Publisher Does Not Support OOB Legal Bases

The CMP reads a TC String from global context storage and it contains a ***DisclosedVendors*** segment:

[***Core***].[***DisclosedVendors***]

```
COvFyGBOvFyGBAbAAAENAPCAAOAAAAAAAAAAAEEUACCKAAA.IFoEUQQgAIQwgIwQABAEAAAAOIAACAIAAAAQ
AIAgEAACEAAAAAgAQBAAAAAAAGBAAgAAAAAAAFAAECAAAgAAQARAEQAAAAAJAAIAAgAAAYQEAAAQmAgBC3ZA
YzUw
```

Because the publisher does not support OOB legal bases, the dot-delimited ***DisclosedVendors*** segment at the end of the TC String is removed when requested form the CMP API:

[***Core***]

```
COvFyGBOvFyGBAbAAAENAPCAAOAAAAAAAAAAAEEUACCKAAA
```

Example 2: A Publisher Supports OOB Legal Bases

The CMP reads a TC String from global context storage and it contains a ***DisclosedVendors*** segment (same as Example 1):

[***Core***].[***DisclosedVendors***]

```
COvFyGBOvFyGBAbAAAENAPCAAOAAAAAAAAAAAEEUACCKAAA.IFoEUQQgAIQwgIwQABAEAAAAOIAACAIAAAAQ
AIAgEAACEAAAAAgAQBAAAAAAAGBAAgAAAAAAAFAAECAAAgAAQARAEQAAAAAJAAIAAgAAAYQEAAAQmAgBC3ZA
YzUw
```

Since the publisher supports OOB legal bases for any vendor that uses it, the TC String, when surfaced through the CMP API, is unchanged from storage – it includes the ***DisclosedVendors*** segment:

[***Core***].[***DisclosedVendors***]

```
COvFyGBOvFyGBAbAAAENAPCAAOAAAAAAAAAAAEEUACCKAAA.IFoEUQQgAIQwgIwQABAEAAAAOIAACAIAAAAQ
AIAgEAACEAAAAAgAQBAAAAAAAGBAAgAAAAAAAFAAECAAAgAAQARAEQAAAAAJAAIAAgAAAYQEAAAQmAgBC3ZA
YzUw
```

Example 3: A Publisher Supports OOB Legal Bases for Only Select Vendors

The CMP reads a TC String from global context storage and it contains a ***DisclosedVendors*** segment (same as Example 1 & Example 2):

[***Core***].[***DisclosedVendors***]

```
COvFyGBOvFyGBAbAAAENAPCAAOAAAAAAAAAAAEEUACCKAAA.IFoEUQQgAIQwgIwQABAEAAAAOIAACAIAAAAQ
AIAgEAACEAAAAAgAQBAAAAAAAGBAAgAAAAAAAFAAECAAAgAAQARAEQAAAAAJAAIAAgAAAYQEAAAQmAgBC3ZA
YzUw.PVAfDObdrA
```

To indicate the select vendors a publisher approves to use OOB legal bases, the CMP includes the ***AllowedVendors*** segment with the TC String from the CMP API:

[***Core***].[***DisclosedVendors***].[***AllowedVendors***]

```
CGL23UdMFJzvuA9ACCENAXCEAC0AAGrAAA5YA5ht7-_d_7_vd-f-
nrf4_4A4hM4JCKoK4YhmAqABgAEgAA.IFut_a83_Ma_t-_SvB3v4-
IAeIAACAIgSAAQAIAgEQACEABAAAgAQFAEAIAAAGBAAgAAAAQAIFAAMCQAAgAAQiRAEQAAAAANAAIAAggAIY
QFAAARmggBC3ZCYzU2yIA.QFulWfTw4obx_Z2zUj6XkNIAeIAACAIgSAAQAIAgEQACEABAAAgAQFAEAIAAAG
BAAgAAAAQAIFAAMCQAAgAAQiRAEQAAAAANAAIAAggAIYQFAAARmggBC3ZCYzU2yIA
```

Disclosed Vendors (OOB)

The ***DisclosedVendors*** is a TC String segment that signals which vendors have been disclosed to a given user by a CMP. This segment is required when saving a global-context TC String. When a CMP updates a globally-scoped TC String, the CMP MUST retain the existing values and only add new disclosed Vendor IDs that had not been added by other CMPs in prior interactions with this user.

Field Name	Bits	Values	Description
SegmentType	3 bits	Enum `0` Default (*Core*) `1` ***DisclosedVendors*** `2` *AllowedVendors* `3` PublisherTC	***DisclosedVendors*** segment is `1` which is `001` in binary.
MaxVendorId	16 bits	The maximum Vendor ID included in this encoding.	Because this section can be a variable length, this indicates the last ID of the section so that a decoder will know when it has reached the end.
IsRangeEncoding	1 bit	`1` Range `0` BitField	The encoding scheme used to encode the IDs in the section – Either a BitField Section or Range Section follows. Encoding logic should choose the encoding scheme that results in the smaller output size for a given set.
BitField Section		**Encodes one disclosed vendor bit per Vendor ID**	
BitField	MaxVendorId bits	One bit for each vendor `1` Disclosed `0` Not Disclosed	The value for each Vendor ID from `1` to MaxVendorId. Set the bit corresponding to a given vendor to `1` if the CMP has disclosed the vendor in the UI.

Range Section		Encodes range groups of Vendor IDs who have been disclosed to a user	
NumEntries	12 bits	Number of ReangeEntry sections to follow	
RangeEntry (repeated NumEntries times)		A single or range of Vendor ID(s) of Vendor(s) who were disclosed in a CMP UI to the user. If a Vendor ID is not within the bounds of the ranges then they were not disclosed to the user.	
IsARange	1 bit	`1` Vendor ID range `0` Single Vendor ID	If more than one Vendor ID is included in this RangeEntry then this describes a range of Vendor IDs and this value is 1. If only one Vendor ID is included then the value is `0`.
StartOrOnlyVendorId	16 bits	The first ID of an inclusive contiguous ascending-order series of Vendor IDs even if the series is only a cardinality of 1.	This is the first or only Vendor ID that has been disclosed in this RangeEntry.
EndVendorId	16 bits	The last ID of the inclusive contiguous ascending-order series of Vendor IDs started with StartOrOnlyVendorId but only if that series has a cardinality greater than 1, otherwise this field is omitted.	The end of the series of Vendor IDs – this is omitted if `IsARange=0`.

Allowed Vendors (OOB)

Signals which vendors the publisher permits to use OOB legal bases.

Field Name	Bits	Values	Description
SegmentType	3 bits	Enum `0` Default (*Core*) `1` *DisclosedVendors* `2` ***AllowedVendors*** `3` PublisherTC	OOB AllowedVendors segment is `2` which is `010` in binary.

MaxVendorId	16 bits	The maximum Vendor ID that is included.	Because this section can be a variable length, this indicates the last ID of the section so that a decoder will know when it has reached the end.
IsRangeEncoding	1 bit	`1` Range `0` BitField	The encoding scheme used to encode the IDs in the section – Either a BitField Section or Range Section follows. Encoding logic should choose the encoding scheme that results in the smaller output size for a given set.
BitField Section		**Encodes one allowed vendor bit per Vendor ID**	
BitField	MaxVendorId bits	One bit for each vendor `1` Allowed `0` Not Allowed	The value for each Vendor ID from `1` to MaxVendorId. Set the bit corresponding to a given Vendor ID to `1` if the Publisher permits the vendor to use OOB legal bases.
Range Section		**Encodes range groups of Vendor IDs who the publisher is allowing to use OOB legal bases**	
NumEntries	12 bits	Number of RangeEntry sections to follow	
RangeEntry (repeated NumEntries times)		A single or range of Vendor ID(s) of Vendor(s) who are allowed to use OOB legal bases on the given publisher's digital property. If a Vendor ID is not within the bounds of the ranges then they are not allowed to use OOB legal bases on the given publisher's digital property..	
IsARange	1 bit	`1` Vendor ID range `0` Single Vendor ID	If more than one Vendor ID is included in this RangeEntry then this describes a range of Vendor IDs and this value is 1. If only one Vendor ID is included then the value is `0`.
		The first ID of an inclusive contiguous	This is the first or only

StartOrOnlyVendorId	16 bits	ascending-order series of Vendor IDs even if the series is only a cardinality of 1.	Vendor ID that has is allowed in this RangeEntry.
EndVendorId	16 bits	The last ID of the inclusive contiguous ascending-order series of Vendor IDs started with StartOrOnlyVendorId but only if that series has a cardinality greater than 1, otherwise this field is omitted.	The end of the series of Vendor IDs – this is omitted if `IsARange=0`.

Publisher Purposes Transparency and Consent

Publishers may need to establish transparency and consent for a set of personal data processing purposes for their own use. For example, a publisher that wants to set a frequency-capping first-party cookie should request user consent for Purpose 1 "Store and/or access information on a device" in jurisdictions where it is required.

The ***Publisher TC*** segment in the TC string represents publisher purposes transparency & consent signals which is different than the other TC String segments; they are used to collect consumer purposes transparency & consent for vendors. This segment supports the standard list of purposes defined by the TCF as well as Custom Purposes defined by the publisher if they so choose.

Field Name	Bits	Values	Description
SegmentType	3 bits	Enum `0` Default (*Core*) `1` *DisclosedVendors* `2` *AllowedVendors* `3` **PublisherTC**	***PublisherTC*** segment is 3 which is `011` in binary.
		One bit for each Purpose:	The user's consent value for each Purpose established on the legal basis of consent, for the publisher

PubPurposesConsent	24 bits	`1` Consent `0` No Consent	The Purposes are numerically identified and published in the Global Vendor List. From left to right, Purpose 1 maps to the `0`th bit, purpose 24 maps to the bit at index 23.
PubPurposesLITransparency	24 bits	One bit for each Purpose: `1` legitimate interest established `0` legitimate interest was **NOT** established or it was established but user exercised their "Right to Object" to the Purpose	The Purpose's transparency requirements are met for each Purpose established on the legal basis of legitimate interest and the user has not exercised their "Right to Object" to that Purpose. By default or if the user has exercised their "Right to Object to a Purpose, the corresponding bit for that purpose is set to `0`. From left to right, Purpose 1 maps to the `0`th bit, purpose 24 maps to the bit at index 23.
NumCustomPurposes	6 bits	The number of Custom Purposes.	Custom purpose IDs are numbered `1` to NumberCustomPurposes. Custom purposes will be defined by the publisher and displayed to a user in a CMP user interface. If the publisher does not use any Custom Purposes, this field is set to `0` and the following two fields will be omitted.
CustomPurposesConsent	NumCustomPurposes	One bit for each Custom Purpose: `1` Consent `0` No Consent	The consent value for each CustomPurposeId from 1 to NumberCustomPurposes

CustomPurposesLITransparency	NumCustomPurposes	One bit for each Custom Purpose: `1` legitimate interest established `0` legitimate interest was **NOT** established or it was established but user exercised their "Right to Object" to the Custom Purpose	The legitimate Interest disclosure establishment value for each CustomPurposeId from `1` to NumberCustomPurposes

The Global Vendor List

The Global Vendor List (GVL) is a technical document that CMPs download from a domain managed and published by IAB Europe. It lists all registered and approved Vendors, as well as standard Purposes, Special Purposes, Features, Special Features and Stacks. The information stored in the GVL is used for determining what legal disclosures must be made to the user.

I'm a vendor, how do I get added to the Global Vendor List?

The registration process is described here: https://iabeurope.eu/tcf

What is contained in the Global Vendor List?

- A Global Vendor List Specification Version
- A Global Vendor List version
- A TCF Policy Version
- A Last Updated Date
- A list of standard Purposes
- A list of Special Purposes
- A list of standard Features
- A list of Special Features.
- A list of Stacks
- A list of Vendors and their:
    - Numeric ID which is incrementally assigned and never re-used – deleted Vendors are just marked as deleted.
    - Name.
    - List of Purposes for which they are requesting consent.

- List of Purposes for which they require to be transparently disclosed as their legitimate interest.
- List of Purposes they have the flexibility to either use a consent or a legitimate interest legal basis.
- List of Special Purposes to transparently disclose as their legitimate interest that a user has no right to object.
- List of Features they use across Purposes.
- List of Special Features they use across Purposes.
- GDPR/privacy policy page URL.
- HTTP "overflow" options which includes a `GET` request maximum size in kilobytes to help diagnose problems with TC String passing as well as limit oversized strings.

Where can I access the Global Vendor List?

The GVL is in JSON format and the current version at any given time can be retrieved using the following URL structure:

https://vendorlist.consensu.org/v2/vendor-list.json

Previous versions of the Global Vendor List are available here:

https://vendorlist.consensu.org/v2/archives/vendor-list-v{vendor-list-version}.json

Where 'vendor-list-version' corresponds to the 'vendorListVersion' property in the GVL, for example, the following URL would retrieve the GVL update published with version 138

https://vendorlist.consensu.org/v2/archives/vendor-list-v138.json

Previous versions of the GVL may only be used in cases when the current version cannot be downloaded (such as when operating in-app while offline), or for change control management.

TCF version 1 of the Global Vendor List (deprecated)

For reference, the URL for version 1 of the TCF was:

https://vendorlist.consensu.org/vendorlist.json

Version 1 of the Global Vendor List and all version 1 archives will continue to be maintained until support officially ends in 2020. At that time, these files will be deprecated and only version 2 and newer of the Global Vendor List will be available.

Translations for Purposes, Special Purposes, Features, and Special Features

Translations of the names and descriptions for Purposes, Special Purposes, Features, and Special Features to non-English languages are contained in a file where attributes containing English content (except vendor declaration information) are translated, and can be found here:

https://vendorlist.consensu.org/v2/purposes-{language}.json

Where 'language' is a two letter lowercase ISO 639-1 language code. Supported languages are listed at the following URL:

https://register.consensu.org/Translation

How often is the Global Vendor List updated?

As of the publication of this document, changes to the Global Vendor List are published weekly at 5:00 PM Central European Time on Thursdays. IAB Europe reserves the right to change this time and will notify CMP members of any changes.

CMPs using the GVL

CMPs must use the **latest** available version of the GVL whenever the interface is surfaced to the user to provide transparency or request consent. This condition applies to first-time and renewal interactions, as well as interactions that involve reviewing and updating settings.

In some cases, due to caching requirements and low connectivity environments, such as for mobile in-app experiences, it may not be possible to use the latest version of the GVL:

- For a delay caused by caching requirements, the penultimate version of the GVL may be considered the latest available version.
- For a delay caused by a lack of connectivity, the last cached version of the GVL may be used but must be updated as soon as connectivity is restored.

To determine whether the interface should be resurfaced to a user, the CMP must compare the latest version of the GVL with the archived version of the GVL identified in the TC String (assuming they are different). The CMP is not required to resurface the interface to the user if the versions are different. The timing and reasons for resurfacing the interface to users is at the discretion of the CMP and the publisher.

Strict restrictions on accessing and caching the GVL apply and are detailed below.

Vendors using the GVL

Vendors must use the version of the GVL encoded in the TC String received to:

- Determine if they have the legal bases they need to process the user's personal data.
- Determine if any vendor they are about to pass personal data to, also has the necessary legal bases to process personal data.

Strict restrictions on accessing and caching the GVL apply and are detailed below.

Accessing and caching the Global Vendor List

In TCF v1 it was possible for client-side CMP applications to load the GVL directly via CORS. Given the scale of the TCF and the high volume of requests for the Global Vendor List, this is no longer possible from TCF v2.0 onward. All requests for the GVL must now be server-side.

Current and previous GVL resources, as well as purpose translations, are configured with cache-control headers. Server-side applications must cache these resources in the same way that a browser would - the file must be requested and cached using the specified `max-age` value in the header. Once the cache expires, the resource can be requested again. Resources must not be cached for a period other than `max-age`.

Previous versions of the GVL must be cached for at least the period specified by the cache-control headers and may be cached indefinitely as they are static resources.

CMPs accessing and caching the GVL

Client-side CMP applications must not load GVL resources directly from vendorlist.consensu.org - instead they must be loaded and hosted by a CMP's server-side application and then passed to the client-side CMP application.

As stated above, CMP server-side applications must cache these resources in the same way that a browser would. For example, if the `max-age` value in the header is one week, the server-side application must do the following:

- Request a GVL resource
- Cache the resource for one week
- When the cache expires after one week, clear the cache if necessary and re-request the resource

Important: The volume of usage will be monitored carefully by the managing organisation and any CMP not adhering to this request limit will be blocked from accessing the GVL.

To prevent client-side applications from repeatedly downloading files, CMPs should set cache-control headers on HTTP responses sent to client-side requests for their self-hosted GVL resources. This will ensure that browsers automatically cache the resources, circumventing the need to repeatedly request files over HTTP.

Vendors accessing and caching the GVL

Vendor requests for GVL resources must be loaded and cached by the server-side application.

As stated above, vendor server-side applications must cache these resources in the same way that a browser would. For example, if the `max-age` value in the header is one week, the server-side application must do the following:

- Request a GVL resource
- Cache the resource for one week
- When the cache expires after one week, clear the cache if necessary and re-request the resource

Important: The volume of usage will be monitored carefully by the managing organisation and any vendor not adhering to this request limit will be blocked from accessing the GVL.

Using a compressed version of the Global Vendor List

In order to control the bandwidth used by requests for the GVL file, vendors and CMPs must request a compressed version of the GVL. This can be done by sending `Accept-Encoding` headers on the `GET` request for the file.

Example:

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

A browser will add this header automatically and, therefore, nothing needs to be done for an in-browser request. Server-side requests are another matter because server software may not decompress the response automatically. Make sure your server requests send the options your service is capable of decoding in your `Accept-Encoding` header.

Global Vendor List and TCF Policy Updates

When a change occurs in the TCF Policies, the update invalidates the previous declarations of vendors listed on the previous version of the GVL. These policy changes happen infrequently, but when they do, a CMP is required to discard the user's current TC String and resurface the user interface to provide new disclosures, capture new consent, and encode a new TC String without migrating any old values over from the old one.

To determine if TCF Policies have changed, CMPs shall compare the ***TcfPolicyVersion*** encoded in a TC String with the ***TcfPolicyVersion*** property in the latest Global Vendor List published by the Managing Organisation – if the values are different then the TCF Policy has changed and a CMP will be required to provide new disclosures, capture new consent, and encode a new TC String.

Example Global Vendor List JSON Object

Here is an annotated example of the GVL's JSON format:

```
{
  "gvlSpecificationVersion": 2,
  "vendorListVersion": 133, // incremented with each published file change
  "tcfPolicyVersion": 2, // The TCF MO will increment this value whenever a GVL
change (such as adding a new Purpose or Feature or a change in Purpose wording)
legally invalidates existing TC Strings and requires CMPs to re-establish
transparency and consent from users. TCF Policy changes should be relatively
infrequent and only occur when necessary to support changes in global mandate. If
the policy version number in the latest GVL is different from the value in your TC
String, then you need to re-establish transparency and consent for that user. A
version 1 format TC String is considered to have a version value of 1.
  "lastUpdated": "2018-05-28T00:00:00Z",
  "purposes": {

    /**
     * Information published for each Purpose
     *
     * "id": number, REQUIRED
     * "name": string, REQUIRED
     * "description": string, REQUIRED
     * "descriptionLegal": string, REQUIRED
     * "consentable": boolean, OPTIONAL, default=true  false means CMPs should never
afford users the means to provide an opt-in consent choice
     * "rightToObject": boolean, OPTIONAL, default=true  false means CMPs should
never afford users the means to exercise a right to object
    */
    "1": {
      "id": 1,
      "name": "Storage and access of information",
      "description": "...",
      "descriptionLegal": "..."
    },
    // ... more purposes from id=2 to id=9 (up to no higher than id=24)
    "10": {
      "id": 10,
      "name": "Develop and improve product",
      "description": "...",
```

```
      "descriptionLegal": "...",
      "consentable": false,
      "rightToObject": false
    }
  },
  "specialPurposes" : {
    "1": {
      "id": 1,
      "name": "Security, Fraud Prevention, Debugging",
      "description": "...",
      "descriptionLegal": "...",
      "consentable": false,
      "rightToObject": false
    },
    "2": {
      "id": 2,
      "name": "Technical ad and content delivery",
      "description": "...",
      "descriptionLegal": "...",
      "consentable": false,
      "rightToObject": false
    }
  },
  "features" : {
    "1": {
      "id": 1,
      "name": "Matching Data to Offline Sources",
      "description": "Combining data from offline sources that were initially
collected in other contexts",
      "descriptionLegal": "..."
    }

  // ... more features from id=2 up to no higher than id=64.

  },

  /**
   * Special features differ from simple features in that CMPs MUST provide
   * users with a means to signal an opt-in choice as to whether vendors
   * may employ the feature when performing any purpose processing.
   * See Policies for specifics.
   */
  "specialFeatures" : {
    "1": {
      "id": 1,
      "name": "Precise Geolocation",
      "description": "...",
      "descriptionLegal": "..."
    },
    "2": {
      "id": 2,
      "name": "Active Fingerprinting",
```

```
    "description": "...",
    "descriptionLegal": "..."
  }

// ... more special features from id=3 up to no higher than id=8.
  //
},
"vendors": {
/**
 * Information published for each vendor
 *
 * "id": numeric, REQUIRED
 *
 * "name": string, REQUIRED
 *
 * "purposes": array of positive integers, either purposes or
 *
 * "legIntPurposes" REQUIRED. Array may be empty. List of purpose ids
 * declared as performed on the legal basis of consent
 *
 * "specialPurposes": array of positive integers, OPTIONAL. Array may be
 * empty. List of Special Purposes declared as performed on the legal basis
 * of a legitimate interest
 *
 * "flexiblePurposes": array of positive integers, OPTIONAL. Array may be
 * empty. List of purpose ids where the vendor is flexible regarding the
 * legal basis; they will perform the processing based on consent or a
 * legitimate interest. The 'default' is determined by which of the other two
 * mutually-exclusive purpose fields is used to declare the purpose for the
 * vendor
 *
 * Constraints:
 *   Either purposes OR legIntPurposes can be missing/empty, but not
 *   both.
 *
 *   A Purpose id must not be present in both purposes and legIntPurposes
 *
 *   A Purpose id listed in flexiblePurposes must have been declared in one
 *   of purposes or legIntPurposes.
 *
 *   Purpose id values included in the three purpose fields must be in the
 *   range from 1 to N, where N is the highest purpose id published in this
 *   GVL file.
 *
 * "features": array of positive integers, OPTIONAL. Array may be empty. List
 * of Features the Vendor may utilize when performing some declared Purposes
 * processing.
 *
 * "specialFeatures": array of positive integers, OPTIONAL. Array may be
 * empty. List of Special Features the Vendor may utilize when performing
 * some declared Purposes processing.
 *
```

```
   * "SpecialPurposes": array of positive integers, OPTIONAL. Array may be
   * empty. List of Special Purposes declared as performed on the legal basis
   * of a legitimate interest
   *
   * "policyUrl": url string, REQUIRED URL to the Vendor's privacy policy
   * document.
   *
   * "deletedDate": date string ("2019-05-28T00:00:00Z") OPTIONAL, If present,
   * vendor is considered deleted after this date/time and MUST NOT be
   * established to users.
   *
   * "overflow": object specifying the vendor's http GET request length  limit
   * OPTIONAL. Has the following members & values
   *
   *   "overflow": {
   *     "httpGetLimit": 32   /* 32 or 128 are supported options */
   *   }
   *    If a vendor entry does not include this attribute then the vendor has no
   *    overflow options and none can be inferred.
   */

  "1":{
    "id": 1,
    "name": "Vendor Name",
    "purposes": [1],
    "specialPurposes": [1],
    "legIntPurposes": [2, 3],
    "flexiblePurposes": [1, 2],
    "features": [1, 2],
    "specialFeatures": [1, 2],
    "policyUrl": "https://vendorname.com/gdpr.html",
    "deletedDate": "2019-02-28T00:00:00Z",
    "overflow": {
      "httpGetLimit": 32   /* 32 or 128 are supported options */
      }
    }
  // ... more vendors
  },

  "stacks": {
    "1": {
      "id": 1,
      "purposes" : [],
      "specialFeatures" : [1,2],
      "name" : "Precise geolocation data, and identification through device
scanning",
      "description" : "Precise geolocation and information about device
characteristics can be used."
    }
  }
}
```

Global CMP List Specification

The Global CMP List (GCL) is a JSON format document that lists all CMPs registered with the Transparency and Consent Framework (TCF). There are separate files for v1.1 and v2 of the framework. These files are used by vendors to determine which CMPs are compliant and active within the framework, in order to ascertain whether a given CMP ID found in a consent string or TC String is valid.

IMPORTANT NOTE: all CMPs that have registered with the TCF are listed in these files. CMPs that are no longer active for whatever reason, have the `deletedDate` property set. Consent strings or TC Strings for CMPs with a `deletedDate` set must be considered invalid after that date/time and must be discarded immediately and not passed downstream.

What is contained in the Global CMP List?

- A Last Updated Date.
- A list of CMPs detailing:
  - A Numeric ID which is incrementally assigned and never re-used - inactive CMPs are marked as deleted.
  - Their Name.
  - Whether or not the CMP is a commercial service.
  - If applicable, the date/time after which CMP is considered inactive.

Where can I access the Global CMP List?

The GCL is in JSON format and the current version at any given time can be retrieved using the following URL:

v2: https://cmplist.consensu.org/v2/cmp-list.json v1: https://cmplist.consensu.org/cmp-list.json

How often is the Global CMP List updated?

As of the publication of this document, changes to the Global CMP List are published weekly at 5:00 PM Central European Time on Thursdays. IAB Europe reserves the right to change this time and will notify members of any changes.

Caching the Global CMP List

Strict restrictions on caching the GCL apply.

All requests for the Global CMP List must honour the cache-control headers and must not cache the resource with different settings.

Note: There may be a delay of up to the maximum cache interval in retrieving the latest version of the Global CMP List.

Server-side caching of the GCL

As requests for a GCL file will not be in a browser context, GCL files must be cached explicitly server-side according to the cache-control headers.

Application logic must only request one version of the GCL during the cache period specified in the cache-control header. For example, if the caching period is one week, only one request for the current GCL file must be received per week.

Note: The volume of usage will be monitored carefully by the managing organisation (MO) and any organisations not adhering to this request limit will be blocked from accessing the GCL.

Using a compressed version of the Global CMP List

A compressed version of the GCL must be requested. This can be done by sending Accept-Encoding headers on the GET request for the file:

- Example: Accept-Encoding: gzip, deflate, br

Example Global CMP List JSON Object

Here is an example of the GCL's JSON format:

```
{
  "lastUpdated": "2019-10-31T00:00:00Z",
  "cmps": {

    /**
     * Information published for each CMP
     *
     * "id": numeric, REQUIRED
     * "name": string, REQUIRED
     * "isCommercial": boolean, REQUIRED
     * "deletedDate": date string ("2019-05-28T00:00:00Z") OPTIONAL
     *  If present, CMP is considered deleted after this date/time and
     *  consent string or TC String must be discarded immediately.
     */

    "2":{
      "id": 2,
      "name": "Chandago",
      "isCommercial": true
    },

    // ... more CMPs

    "136":{
      "id": 136,
      "name": "M6 Web",
      "isCommercial": false,
      "deletedDate": "2019-08-06T00:00:00Z"
    }

    // ... more CMPs

  }
}
```