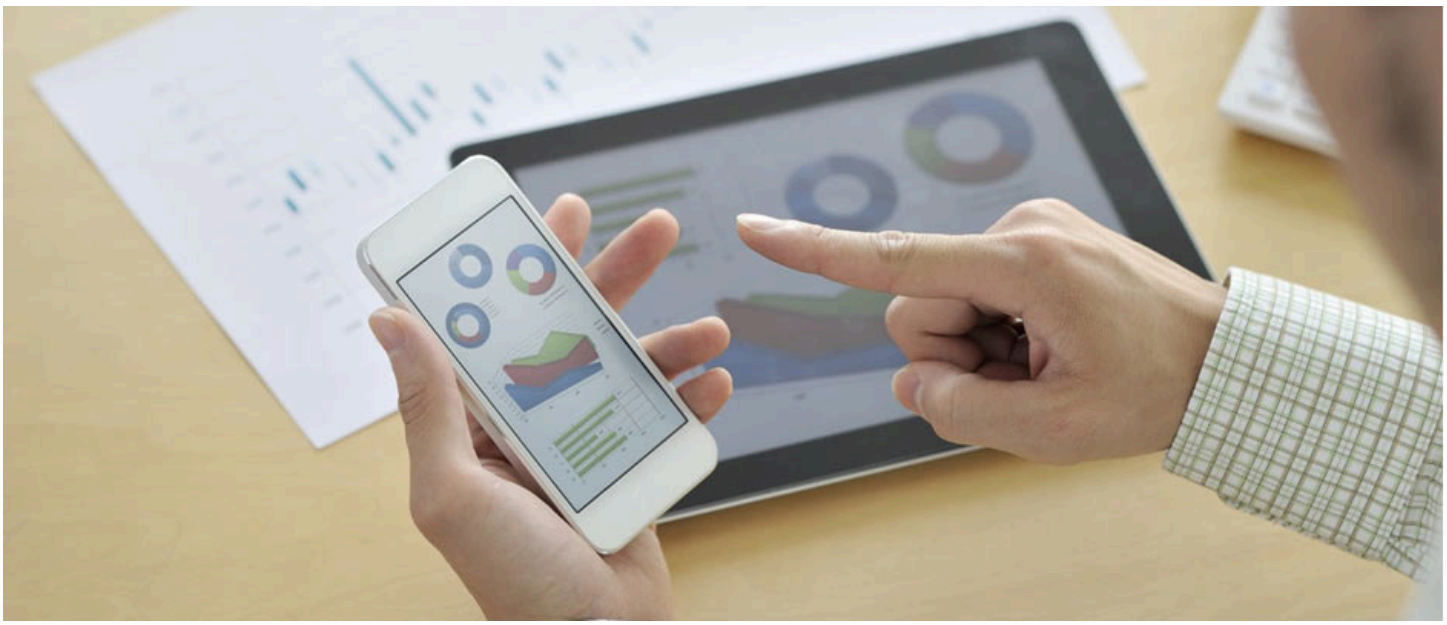
scientiamobile

*"The first step in a great mobile experience"*



# REQUEST AND IMPLEMENT USER-AGENT CLIENT HINTS

## Support

The ScientiaMobile Enterprise Support Portal is open to all WURFL users, both commercial license holders and evaluation users. It represents the combined knowledge base for the WURFL community. Commercial licensees are invited to post questions in the forum using the account to which their licenses are associated. This may mean faster handling of those posts by ScientiaMobile's personnel.

For commercial license holders, there are tiered support levels to address a variety of business support needs. After logging into your account, commercial licensees with support options can access the Enterprise Support portal to post tickets. These tickets will receive expedited attention.

To inquire about support plans, use our License Inquiry or our General Inquiry form.

## Update Notifications

If you would like to be notified of our API updates, major data updates, and other technical changes, please subscribe to our ScientiaMobile Announcements list.

scientiamobile

www.scientiamobile.com

Tel +1.703.310.6650

E-mail: sales@scientiamobile.com

Copyright © 2026 ScientiaMobile, all rights reserved. WURFL Cloud, WURFL OnSite, WURFL and, InFuze WURFL InSight and respective logos are trademarks of ScientiaMobile. Apache is the trademark of the Apache Software Foundation. NGINX is the trademark of Nginx Software Inc. Varnish is the trademark of Varnish Software AB

# Request and Implement User-Agent Client Hints

# Introduction

ScientiaMobile has been following the evolution of User-Agent Client-Hints closely and we have already written a few articles that talk about the **plans for User-Agent Client-Hints** and **support for them in the WURFL API**. ImageEngine has a **section dedicated to enabling Client Hint support**. Google has a few **developer resources** that **talk about this** as well.

In the conversations we have had with our customers and partners over the past two years regarding User-Agent Client-Hints, one question kept popping up - how do I as a website owner start receiving User-Agent Client-Hints and what is the best way to pass this information to WURFL?

# Low entropy and high entropy User-Agent Client-Hints

User-Agent Client-Hints can be broadly divided into two tiers - those that are sent with every request (Low Entropy Client Hints) and those that must specifically be requested by the server (High Entropy Client Hints). Below is a table of all the User-Agent Client-Hints (at the time of writing) and how they are sent.

| User-Agent Client-Hint | Low/High Entropy |
|---|---|
| **sec-ch-ua** | Low Entropy |
| **sec-ch-ua-mobile** | Low Entropy |
| **sec-ch-ua-platform** | Low Entropy |
| **sec-ch-ua-platform-version** | High Entropy |
| **sec-ch-ua-full-version (deprecated)** | High Entropy |
| **sec-ch-ua-full-version-list** | High Entropy |
| **sec-ch-ua-model** | High Entropy |
| **sec-ch-ua-arch** | High Entropy |
| **sec-ch-ua-bitness** | High Entropy |
| **sec-ch-ua-wow64** | High Entropy |

# Opt in to receive User-Agent Client-Hints

Your server's response headers must specifically advertise support for User-Agent Client-Hints using the **Accept-CH response header**. This response header must contain *all* high entropy User-Agent Client-Hints that you need. Here is

an example Accept-CH header:

```
Accept-CH: sec-ch-ua-platform-version,sec-ch-ua-full-version,sec-ch-ua-full-version-list,sec-ch-ua-model,sec-ch-ua-arch,sec-ch-ua-bitness,sec-ch-ua-wow64
```

Previously, you could use “Accept-CH” as a meta attribute in your HTML:

```
<meta http-equiv="Accept-CH" content="sec-ch-ua-platform-version,sec-ch-ua-full-version,sec-ch-ua-full-version-list,sec-ch-ua-model,sec-ch-ua-arch,sec-ch-ua-bitness,sec-ch-ua-wow64" />
```

However, the Accept-CH meta tag is **now** considered to be a legacy feature and the use of the "**Delegate-CH**" meta tag is encouraged in its place. The "Delegate-CH" meta tag allows first party sub-request and third party delegation of high entropy Client Hints in a single tag:

```
<meta http-equiv="delegate-ch" content="sec-ch-ua https://wurfl.io
https://wjs.wurflcloud.com; sec-ch-ua-bitness https://wurfl.io
https://wjs.wurflcloud.com; sec-ch-ua-arch https://wurfl.io
https://wjs.wurflcloud.com; sec-ch-ua-model https://wurfl.io
https://wjs.wurflcloud.com; sec-ch-ua-platform https://wurfl.io
https://wjs.wurflcloud.com; sec-ch-ua-platform-version https://wurfl.io
https://wjs.wurflcloud.com; sec-ch-ua-full-version https://wurfl.io
https://wjs.wurflcloud.com; sec-ch-ua-full-version-list https://wurfl.io
https://wjs.wurflcloud.com; sec-ch-ua-mobile https://wurfl.io
https://wjs.wurflcloud.com">
```

Note: When using the "Delegate-CH" meta tag, the document opting in will not receive any high entropy Client Hints itself. However, any third parties that are explicitly listed in the tag and any first party sub-requests/requests to other assets in the same origin will have access to high entropy Client Hints. Please also note that, as currently designed, the headers listed in the "Delegate-CH" meta tag are not added to the **Accept-CH cache**.

# Get User-Agent Client-Hints quicker (optional but highly recommended)

When your server advertises support for User-Agent Client-Hints using the Accept-CH response header, the client device will only start sending the High Entropy User-Agent Client-Hints with the second request. This means that the very first request that the client sends will not have all the information you need and that you will need to somehow link the data from the first and second requests together to avoid duplicating visitor data.

This need for a second request can somewhat be mitigated by using one of two mechanisms: the **Critical-CH response header** and **Application-Layer Protocol Settings (ALPS)**. Here is an example Critical-CH header:

```
Critical-CH: sec-ch-ua-platform-version,sec-ch-ua-full-version,sec-ch-ua-full-version-list,sec-ch-ua-model,sec-ch-ua-arch,sec-ch-ua-bitness,sec-ch-ua-wow64
```

# Delegate access to the User-Agent Client-Hints data using a permissions policy (for WURFL.js and ImageEngine customers)

*If you are using the "Delegate-CH" meta tag as discussed above, high entropy Client Hint delegation for third parties is already taken care of and this section can be safely skipped. If you are using the "Accept-CH" header and are not setting the "Delegate-CH" meta tag, please continue to read this section.*

Now that you have requested the client to send User-Agent Client-Hints, you will need to share that data with us so that device detection can be performed. **Permissions policy** (formerly called feature policy) is a mechanism where you give third party websites selective access to browser data. The permissions policy is a **structured header** with the values containing the browser data and the websites that have access to this data. For example, to share User-Agent Client-Hints data with **WURFL.js** Lite, you would implement this permissions policy:

```
permissions-policy: ch-ua-platform-version=(self "https://wurfl.io"),ch-ua-full-version=(self "https://wurfl.io"),ch-ua-full-version-list=(self "https://wurfl.io"),ch-ua-model=(self "https://wurfl.io"),ch-ua-arch=(self "https://wurfl.io"),ch-ua-bitness=(self "https://wurfl.io"),ch-ua-wow64=(self "https://wurfl.io")
```

Note: Adding a `self` delegation to the permissions policy allows first party sub-requests to access the high entropy User-Agent Client Hints as well.

WURFL.js Basic, Standard and Pro customers should delegate access to "*https://wjs.wurflcloud.com*" instead. ImageEngine customers should refer to **this document** for more information.

# User-Agent Reduction Deprecation Origin Trial

**There is a proposal** to create an origin trial for websites who require additional time to migrate to utilizing User-Agent Client-Hints. This origin trial will allow websites to access the non-reduced form of the User-Agent via the standard User-Agent header field. Signing up for this origin trial will provide a token that the website will need to include with its response. In addition to the token, the website will need to advertise support for the `Sec-CH-UA-Full` header using the `Accept-CH` mechanism as demonstrated below.

```
Accept-CH: sec-ch-ua-full
```

In addition, the website may also choose to send send a Critical-CH response header or use **ALPS** to get the non-reduced User-Agent string with the initial request. Here is an example Critical-CH header.

```
Critical-CH: sec-ch-ua-full
```

WURFL.js and ImageEngine customers should make sure to additionally delegate access for the `Sec-CH-UA-Full` header to us, using permissions policy. Here's an example permissions policy for **WURFL.js Lite**:

```
permissions-policy: ch-ua-platform-version=("https://wurfl.io"),ch-ua-full-version=
("https://wurfl.io"),ch-ua-full-version-list=("https://wurfl.io"),ch-ua-model=
("https://wurfl.io"),ch-ua-arch=("https://wurfl.io"),ch-ua-bitness=
("https://wurfl.io"),ch-ua-wow64=("https://wurfl.io"),ch-ua-full=("https://wurfl.io")
```

Please note that this origin trial is a proposal and the specific client hints and mechanisms may change prior to implementation. The evolution of this proposal may be monitored **here**.

# What User-Agent Client-Hints does WURFL need?

WURFL has always utilized all information available to it, whether from a User-Agent string or from an HTTP request. WURFL can utilize all HTTP header fields to provide the most accurate device detection possible. In fact, this is what puts us in a unique position of being able to track and adapt to changes in the User-Agent Client-Hints specification so quickly. For best detection results, we recommend that you pass the entire intact HTTP header with the following User-Agent Client-Hints to WURFL:

```
sec-ch-ua,sec-ch-ua-mobile,sec-ch-ua-platform,sec-ch-ua-platform-version,sec-ch-ua-
full-version,sec-ch-ua-full-version-list,sec-ch-ua-model,sec-ch-ua-arch,sec-ch-ua-
bitness,sec-ch-ua-wow64
```

As the User-Agent Client-Hints evolve, we will update our recommendation to include all the hints needed for an accurate device detection.

# Summary

This document covers a few different methods to request, delegate and access User-Agent Client Hints. Here is a reference matrix that shows all previously discussed ways of requesting and delegating User-Agent Client Hints and how this data is made available to first and third parties:

| Client Hint request type | Document/First Party | First Party Sub-requests/Same Origin | Third Party |
|---|---|---|---|
| None | Low Entropy | Low Entropy | Low Entropy |

| Client Hint request type | Document/First Party | First Party Sub-requests/Same Origin | Third Party |
|---|---|---|---|
| Accept-CH Header | Low + High Entropy | Low + High Entropy | Low Entropy |
| Delegate-CH Meta Tag | Low Entropy | Low + High Entropy | Low + High Entropy |
| Permissions Policy Header | Low Entropy | Low Entropy | Low Entropy |
| Accept-CH Header + Delegate-CH Meta Tag | Low + High Entropy | Low + High Entropy | Low + High Entropy |
| Accept-CH Header + Permissions Policy Header (with self) | Low + High Entropy | Low + High Entropy | Low + High Entropy |
| Accept-CH Header + Permissions Policy Header (without self) | Low + High Entropy | Low Entropy | Low + High Entropy |