Supply Chain Object For Non-OpenRTB Requests

This document describes the standard way to communicate Supply Chain information via a tag rather than OpenRTB. You most commonly need to do this when an advertising system provides a tag to be inserted into an ad server, video player, SSAI vendor, and so on, to initiate an ad request.

Goals of the serialization

- Support all properties of Supply Chain object.
- Minimize the need to serialize URL encoding.
- Maintain compatibility for future Supply Chain versions.

Use cases

This appendix outlines a methodology defining well-structured key-value pair Supply Chain data.

A target advertising system handling a tag-based request and forming an outbound OpenRTB request to another advertising system should initiate the following procedure:

1. Parse the incoming Supply Chain data as described below.
2. Create and fill the Supply Chain object with the incoming data.
3. Append target advertising system's own information to the array in the Supply Chain object

Advertising system receives Supply Chain via tags or URLs

The target system should support a parameter in its ad tags or VAST URLs to accept a Supply Chain serialized string. The best practice is to name this parameter, `sChain`. For example, an ad server may have a display ad tag format like:

```
<script src="https://ads.exchange1.com/srv?pid=194&sz=300x250&plid=2842181&sChain=[SUPPLYCHAIN
GOES HERE]"></script>
```

Or a VAST URL format like:

```
https://ads.exchange1.com/srv?pid=194&sz=v&plid=2842185&sChain=[SCHAIN GOES HERE]
```

OpenRTB Supply Chain object serialization within a URL parameter

The best practice is to use the parameter name, `sChain`.

Serialization format

Serialization is composed of the `SChainObject` properties and the `SChainNode` array, separated by a bang (**`!`**) character, like so:

```
{SChainObject}!{SChainNode array}
```

SChainObject properties

`SChainObject` contains two properties: `ver` (version) and `complete`. Separate the two values with a comma (**`,`**), and they must come at the beginning of the serialized value:

```
ver,complete
```

Array of SChainNode properties

Following the `SChainObject` properties, the serialized values must include every node in `SChainNode`. Separate `SChainNode` object properties using commas (**`,`**) and separate multiple nodes bang (**`!`**) characters. Order properties like so:

```
asi,sid,hp,rid,name,domain,ext
```

The `ext` property content is exchange-specific, so is beyond the scope of this document. Consult developer documentation for the target exchange.

You can omit optional `SChainNode` property values, and you can also exclude trailing separators; for example,

> *exampleexchange.com,12345,1,,,*

or

> *exampleexchange.com,12345,1*

URL encode any values that require it, including any values that contain a comma or a bang character.

> **Do not encode…**
>
> Any comma or bang that is a delimiter between values.

For example:

> *exampleexchange.com,123%2CB,1,,,*

This represents an `sid` of "`123,B`" on `exampleexchange.com`, which handles payments.

Error Handling

If the target advertising system of an incoming Supply Chain object is unable to parse the `sChain` data and attempts to pass `sChain` to a downstream recipient, the best practice is to initiate a new `sChain` where `complete=0` rather than trying to interpret the unreadable format.

The best practice for entries that parse but are deemed *invalid* by the advertising system parser, is to include them in the outgoing downstream requests.

Examples

The following examples illustrate the various use cases above.

Single Hop - Complete Chain

```
"sChain":{
  "ver":"1.0",
  "complete":1,
  "nodes":[{
    "asi":"exchange1.com",
    "sid":"1234",
    "hp":1,
    "rid":"bid-request-1",
    "name":"publisher",
    "domain":"publisher.com"
  }]
}
```

Serialized Value

> *1.0,1!exchange1.com,1234,1,bid-request-1,publisher,publisher.com*

Single Hop - Chain Complete, optional fields missing SChain

```
"sChain": {
  "ver":"1.0",
  "complete":1,
  "nodes" : [{
    "asi":"exchange1.com",
    "sid":"1234",
    "hp":1
  }]
}
```

Serialized Value

> *1.0,1!exchange1.com,1234,1,,,*

Multiple Hops - With all properties supplied SChain

```
"sChain": {
  "ver": "1.0",
  "complete" : 1,
  "nodes" : [{
    "asi":"exchange1.com",
    "sid":"1234",
    "hp":1,
    "rid":"bid-request-1",
    "name":"publisher",
    "domain":"publisher.com"
  }, {
    "asi":"exchange2.com",
    "sid":"abcd",
    "hp":1,
    "rid":"bid-request-2",
    "name":"intermediary",
    "domain":"intermediary.com"
  }]
}
```

Serialized Value

> *1.0,1!exchange1.com,1234,1,bid-request-1,publisher,publisher.com!exchange2.com,abcd,1,bid-request2,intermediary,intermediary.com*

Multiple Hops - Chain Complete, optional fields missing

```
"sChain" : {
    "ver":"1.0",
    "complete":1,
    "nodes":[
        {
            "asi":"exchange1.com",
            "sid":"1234",
            "hp":1
}, {
"asi":"exchange2.com",
"sid":"abcd",
"hp":1
} ]
}
```

Serialized Value

> *1.0,1!exchange1.com,1234,1,,,!exchange2.com,abcd,1,,,*

Multiple Hops Expected - Chain Incomplete SChain

```
"sChain" : {
    "ver":"1.0",
    "complete" :0,
    "nodes":[
        {
            "asi":"exchange2.com",
            "sid":"abcd",
            "hp":1
} ]
"asi":"exchange2.com",
"sid":"abcd",
"hp":1
}
```

Serialized Value

> *1.0,0!exchange2.com,abcd,1,,,*

Single Hop - Chain Complete, encoded values SChain

```
"sChain" : {
    "ver":"1.0",
    "complete":1,
    "nodes":[
        {
            "asi":"exchange1.com",
            "sid":"1234!abcd",
            "hp":1,
            "rid":"bid-request-1",
            "name":"publisher, Inc.",
            "domain":"publisher.com"
} ]
}
```

Serialized Value

> *1.0,1!exchange1.com,1234%21abcd,1,bid-request-1,publisher%2c%20Inc.,publisher.com*