Decision Manager for DFP Technical Integration Guide

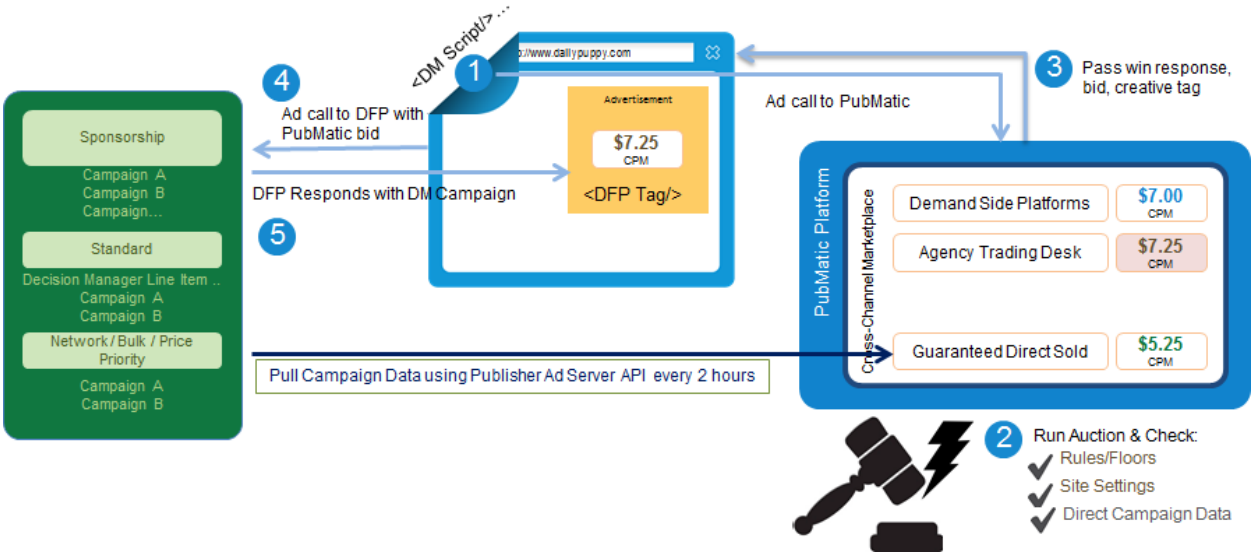
Overview

A publisher's guaranteed and non-guaranteed campaigns typically execute in silos. The general assumption is that non-guaranteed buyers pay a lower price, which is reflected in the priority and non-guaranteed demand they receive in the publisher's ad server. As we see more advertising budgets shift to programmatic channels, there is a need for the publisher to evaluate all available demand in order to allow for true optimization. Decision Manager (DM) empowers publishers to achieve this by comparing pricing for each impression across all channels before the ad server makes the decision.

Below are main benefits of using the Decision Manager Solution:

- Increases revenue by capturing extra advertising dollars from the high-value spot demand obtained from non-guaranteed marketplace
- Real time guidance for publishers' direct sales strategy, as publishers are able to see the programmatic demand for each impression

Decision Manager Workflow



1. Decision Manager's JavaScript code that is placed on the publisher's site, makes an ad call to PubMatic before their ad server. It is a single ad call covering all the placements on the page.
2. PubMatic sends back a bid response from Open and Private RTB marketplaces for all the placements on the page.
   i) The Campaign data present in the publisher's ad server (priority, pacing and targeting), is taken into account while evaluating the impression for optimization and determining the price floor for programmatic marketplace.
   ii) PubMatic Deal Management and Brand Control rules are also taken into account in the auction, in an effort to drive higher value and maintain brand standards.
3. PubMatic's bid response is sent to the publisher's ad server by dynamically adding this response as a key-value to the existing ad server tags. PubMatic DM line items are created in the publisher's ad server at various priorities targeting the bid response from PubMatic.
4. Publisher's ad server makes the final decision on which campaign/line item to serve with PubMatic's bid response as an additional parameter. Lastly, the ad is served.

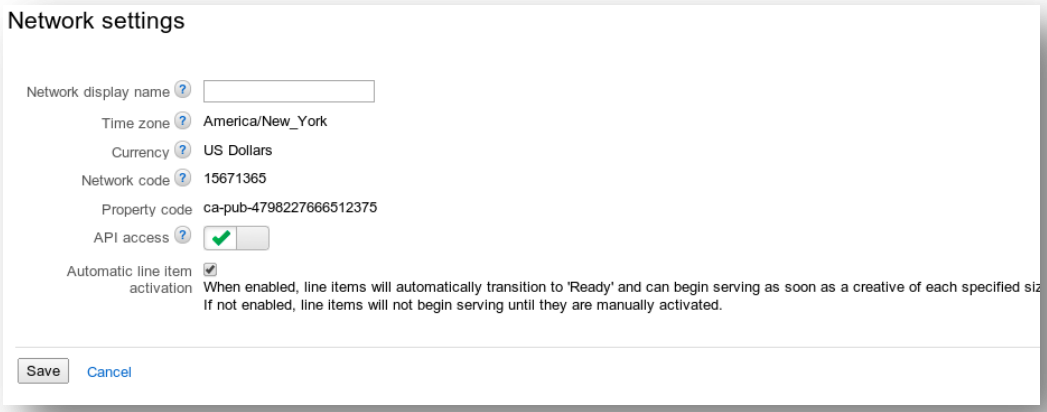
Decision Manager Implementation

Decision Manager provides the pricing from Open and Private marketplaces for ad slots which publishers can consume by passing key-value pairs to the ad server in real time.

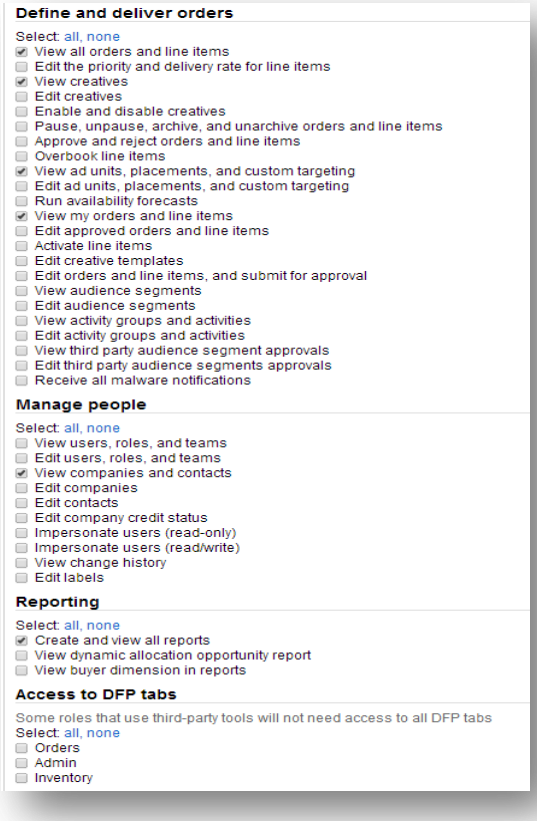
Integrating with Publisher Ad Server

Decision Manager needs to be integrated with the publisher's ad server so that it can use the direct campaigns' CPM and pacing to make the right decisions.

1. Enabling API access
   a. Enable Network level API access:



b. Publisher provides read-only API access to Email ID (new Gmail ID is recommended) provided by PubMatic with following basic permissions



:

2. Provide the network ID for your DFP account to PubMatic in order to complete the integration.

Placing Decision Manager Script on the Page

There are two methods by which you can place the Decision Manager script on your site:

Integration Options	Advantages	Disadvantages
Option 1	• Simple and quick deployment • No need to specify any ad slots' names • No additional code required for consuming the bid prices or targeting parameters sent by PubMatic	• Applicable only for DFP GPT • Integration cannot be customized
Option 2	• Applicable for both Legacy DART and DFP GPT • Integration can be customized	• Requires more time for additional development • Need to specify the ad slots' names • Additional code required for consuming the bid prices or targeting parameters sent by PubMatic

Option 1

The following section is applicable only for **DFP GPT** implementation. In this option, you do not need to develop any additional code for consuming the bid prices or targeting parameters sent by PubMatic.

Note: Refer to the following demo page implemented with the below-mentioned DM script - http://demo.pubmatic.com/uoe/dm_gpt_generic_demo.html. For login details, please contact your PubMatic Account Manager.

To ensure that the DM script loads before GPT, it is recommended to load "gpt.js" with 50 ms delay (standard ping time for Akamai CDN) after the DM script calls are initiated.

For example,

```
setTimeout(function(){
        var gads = document.createElement('script');
        gads.async = true;
        gads.type = 'text/javascript';
        var useSSL = 'https:' == document.location.protocol;
        gads.src = (useSSL ? 'https:' : 'http:') +
        '//www.googletagservices.com/tag/js/gpt.js';
        var node = document.getElementsByTagName('script')[0];
        node.parentNode.insertBefore(gads, node);
}, 50);
```

Decision Manager's multi-slot tag is a JavaScript code, which loads as "dm_gpt_generic.js". This JavaScript reads ad slot names and the publisher ID while querying the PubMatic ad server for the price of the ad slots.

Decision Manager's multi-slot tag is a JavaScript code, which loads as "dm_gpt_generic.js". This JavaScript reads ad slot names and the publisher ID while querying the PubMatic ad server for the price of the ad slots.

The publisher needs to replace the highlighted text in the below script and host it on their site.

```
<script type='text/javascript'>
   var pm_pub_id = 9999; // PubMatic Publisher ID.
   var pm_dm_enabled = true; // For DM implementation.
   var pm_timeout =500; // optional timeout for PubMatic call.

      (function(){
            var pmads = document.createElement('script');
            pmads.async = true;
            pmads.type = 'text/javascript';
            pmads.src = '//ads.pubmatic.com/AdServer/js/dm_gpt_generic.js';
            var node = document.getElementsByTagName('script')[0];
            node.parentNode.insertBefore(pmads, node);
         })();
</script>
```

We recommend deploying it in the head section of your HTML page before the GPT code. It's a special tag which supports getting the price for multiple ad slots on the page using a single ad request. You don't need to deploy separate Decision Manager tags for each ad slot on your page.

This script performs the following actions:

- Defining the slots based on the GPT ad units
- Making the call to PubMatic for the bids
- Adding PubMatic bid's details as targeting to the GPT call.

Option 2

The following section is applicable for **Legacy DART** and **DFP GPT** implementations. In this option, you need to develop additional code for consuming the bid prices and targeting parameters sent by PubMatic.

Note: Refer to the following demo page implemented with the below-mentioned DM script. For login details, please contact your PubMatic Account Manager.

- DART legacy tags - http://demo.pubmatic.com/uoe/uoe_grpbid_gpt_async_demo.html.
- GPT tags with synchronous implementation - http://demo.pubmatic.com/uoe/uoe_grpbid_gpt_sync_demo.html
- GPT tags with asynchronous implementation - http://demo.pubmatic.com/uoe/uoe_grpbid_async_callback_gpt_demo.html
- DART legacy iFrame tags - http://demo.pubmatic.com/uoe/uoe_dart_demo.html

1. Decision Manager's multi-slot tag is a JavaScript code, which loads as "gshowad.js". This JavaScript reads ad slot names and the publisher ID while querying the PubMatic ad server for the price of the ad slots.
   The publisher needs to replace the highlighted text in the below script and host it on their site. The ad slots referred in the below script indicates the ad units' names in DFP.

```
<script type="text/javascript">

// Specify multiple slot names over here
   var pm_optimize_adslots = ['site2_section2@300x250','site3_section3@300x250'];

// PubMatic platform specific publisher identifier
   var pm_pub_id = 99999;

// pm_custom is an optional parameter. Pass the key-values applicable for the impression in it to the ad server
   var pm_custom = "age=25;interest=music,car;gender=male";

</script>

<script type="text/javascript" src=" http://ads.pubmatic.com/AdServer/js/gshowad.js">
</script>
```

We recommend deploying it in the head section of your HTML page. It's a special tag which supports getting the price for multiple ad slots on the page using a single ad request. You don't need to deploy separate Decision Manager tags for each ad slot on your page.

This reduces the overall number of web requests, thereby making page loads faster and reducing the latency.

2. The Decision Manager tag returns the response in the following format:
   For Legacy DART:

```
var progKeyValueMap=
{
'site2_section2@300x250': 'bidstatus=0;bid=2.2;bidid=site2_section2@300x250',

'site3_section3@300x250': 'bidstatus=1;bid=7;bidid=site3_section3@300x250;wdeal=Deal1'
};
```

For DFP GPT:

```
var progKeyValueMap=
{
'site2_section2@300x250': 'bidstatus;0;bid;2.2;bidid;site2_section2@300x250',
'site3_section3@300x250': 'bidstatus;1;bid;7;bidid;site3_section3@300x250;wdeal;Deal1'
};
```

Two global JavaScript variables – 'bidDetailsMap' and 'progKeyValueMap' will be available after the response.

- 'progKeyValueMap' – It contains the slot names as a key and comma-separated list of 3 key-value pairs.
  - 'bidstatus' defines whether PubMatic is able to beat the publisher's ad server's campaign prices or any other specific floor set by the publisher. "bidstatus=1" indicates that PubMatic should serve ahead of guaranteed campaigns. "bidstatus=2" indicates that PubMatic should serve ahead of other non-guaranteed campaigns but below guaranteed campaigns. "bidstatus=3" indicates that PubMatic should serve at the lowest priority.
  - 'bid' identifies the price given by PubMatic for the specific ad slot.
  - 'bidid' is the same as the ad slot's name.
  - 'wdeal' indicates the ID of a PubMatic Marketplace deal, which has provided the winning price. If a deal does not win the impression, then this field is empty.
- 'bidDetailsMap' – This is specifically used by PubMatic while rendering creatives.

3. Publishers can write their own code to consume prices provided by PubMatic to make the decision of whether to serve the PubMatic campaign or pass appropriate key-values to the ad server.

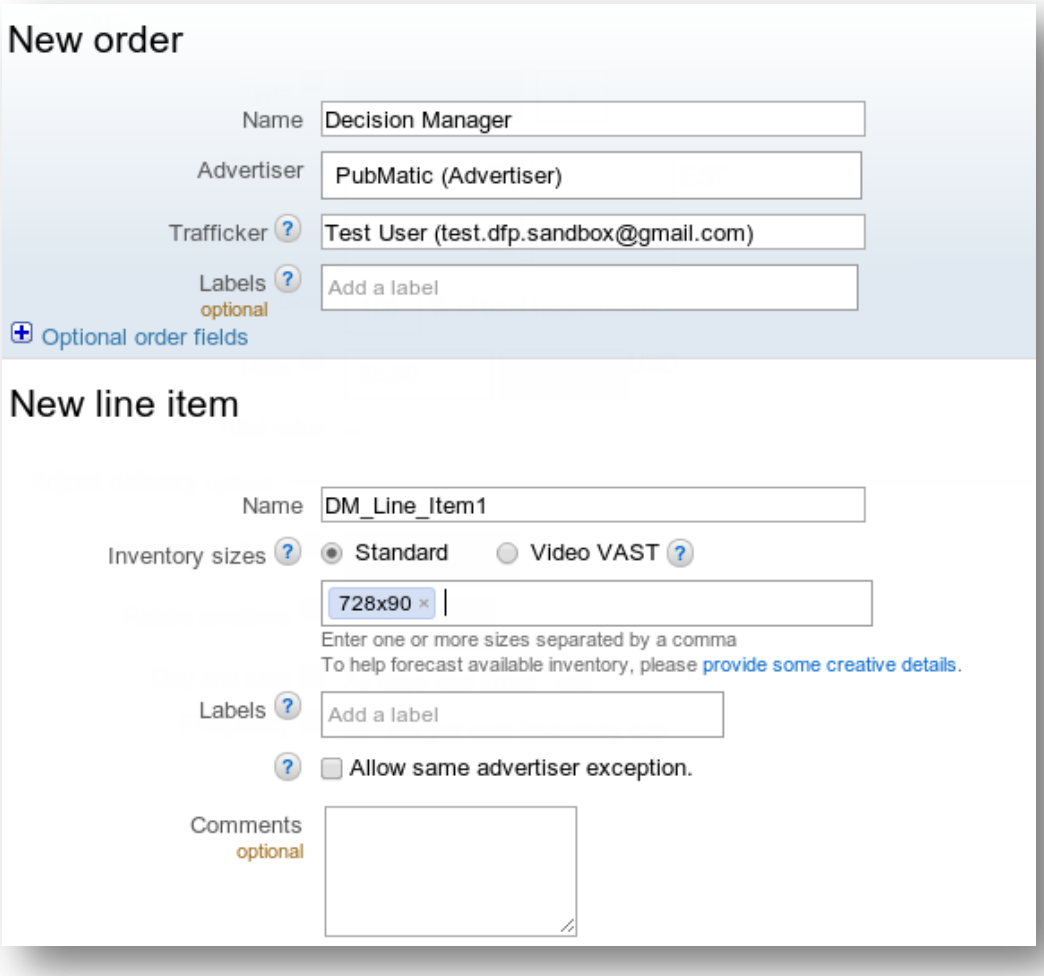
```
if(typeof progKeyValueMap!='undefined'){
var targeting=progKeyValueMap['site2_section2@728x90'].split(";");

<!-- Set key value targeting[0] returns key i.e "bidstatus" targeting [1] returns value that is
"0/1/2/3" similarly targeting[2] and targeting[3] return bid and corresponding value, targeting[4]
and targeting[5] represent bidid -->

s1.setTargeting(targeting[4],targeting[5]);
s1.setTargeting(targeting[0],targeting[1]);
s1.setTargeting(targeting[2],targeting[3]);
}
```

4. When the publisher's ad server selects the PubMatic campaign, it returns a small JavaScript code to include the global variable 'bidDetailsMap' for rendering the actual creative.

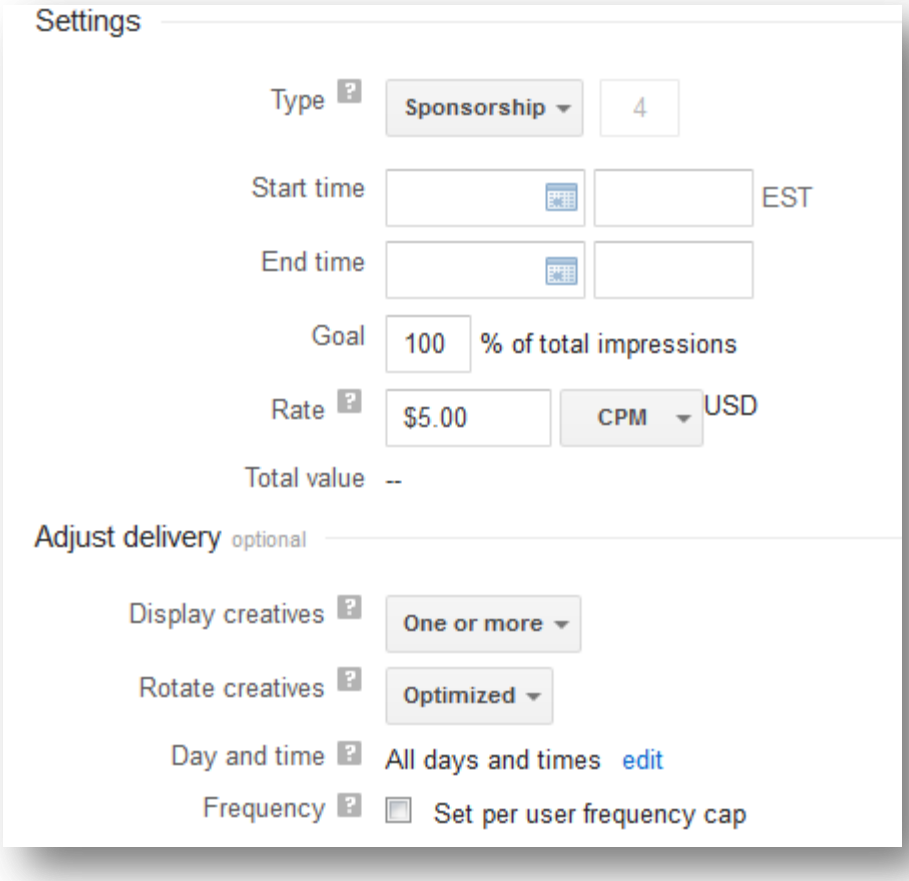
Trafficking Decision Manager Campaign in Ad Server

This step is required so that impressions can be allocated to PubMatic by the ad server based on actual bid price and recommendation from PubMatic. These are passed to the ad server using key-value pairs added to the ad server tags dynamically, as explained in the sections above.

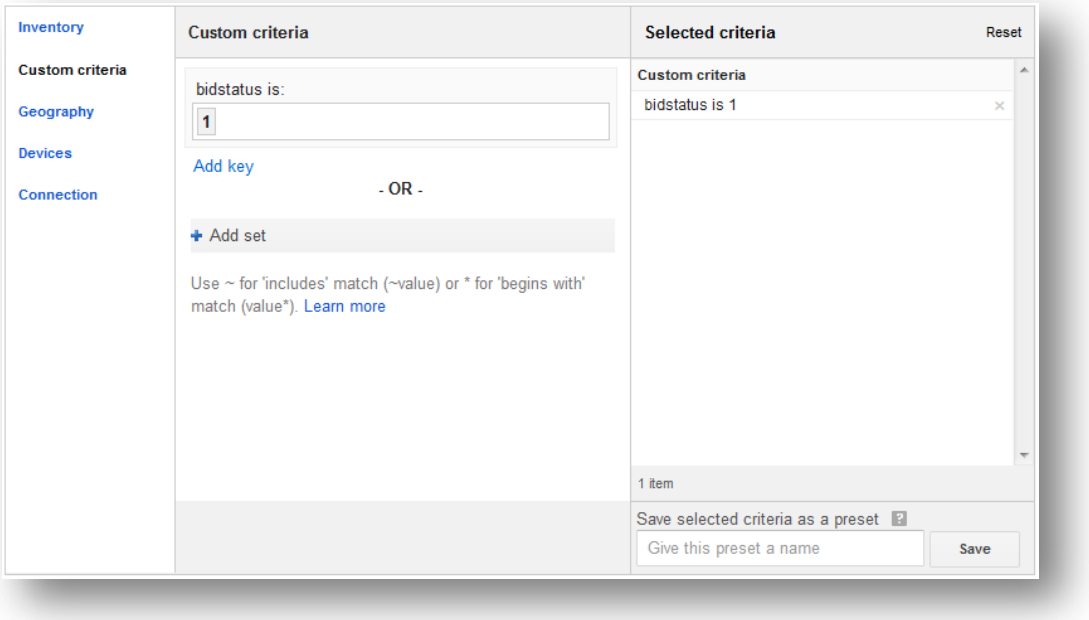
a. Create a new order in DFP for Decision Manager with PubMatic as the advertiser and add the relevant details.



b. Set the price and priority of the line items. You need to create three line items for Decision Manager as given below. Ensure that you also set the goal for each of these line items at an appropriate value; it is recommended to set it at 100%
   - For "bidstatus=1", the line item's type should be "Sponsorship" and priority should be Low.
   - For "bidstatus=2", the line item's type should be "Network" and priority should be above other non-guaranteed line items but below guaranteed line items, that is, High. The "Rate" field should be typically set at a value that is higher than other non-guaranteed line items
   - For "bidstatus=3", the line item's type should be "Network" and priority should be Low, that is, below all non-guaranteed line items. The "Rate" field should be typically set at a value that is lower than other non-guaranteed line items.



c. Set targeting on “bidstatus” value from the DM tag’s response in each line item as described in the previous step.



d. Add the DM creative provided by PubMatic to the line item created.

- **For Option 1-based integration:**

```
<script type="text/javascript">
if(!window.DM){ document.write('<script type="text/javascript" src="'+ ( location.protocol === "https:" ? "https:" : "http:" ) +'//ads.pubmatic.com/AdServer/js/dm_gpt_generic.js"><\/script>');}
</script>
<script type="text/javascript">
DM.displayCreative("%%PATTERN:bidid%%");
</script>
```

- **For Option 2-based integration:**

```
<script type="text/javascript">
if(!window.PubMaticGrouped){ document.write('<script type="text/javascript" src="'+ (
location.protocol === "https:" ? "https:" : "http:" )
+'//ads.pubmatic.com/AdServer/js/gshowad.js"><\/script>');}
</script>
<script type="text/javascript">
PubMaticGrouped.displayCreative("%%PATTERN:bidid%%");
</script>
```