MARKETING INTEGRATION

Best practice guide
Introduction

Looking for ways to effectively scale your marketing on Facebook? This guide will take you through our marketing integration best practices—from the resources recommended for an integration, to the solutions we tailored to help surmount your day-to-day advertising workflow challenges. When you create and manage ads, or extract insights using our Marketing API, you can give your business a competitive advantage by automating your advertising workflows and enabling higher productivity per marketer.

Key benefits of a marketing API integration:

1. **Workflow automation**
   Automatically create and manage ads, test creative combinations and re-allocate budget based on custom configured rules.

2. **Measurement and analytics**
   Pull ads insights at your desired cadence to monitor your ad campaign performance and optimize your ads based on real-time data.

3. **Advanced audience management**
   Plug in your first party data from your CRM to automatically create precise audience segments to re-engage your customers and prospect new ones.

Resourcing recommendations

To build a basic, functioning Marketing API integration, we recommend the following staff for an optimal integration:

**2 engineers**
- To integrate new features for automated ads management, creation, and insights
- Resolve mandatory breaking changes approximately every 4 months
- Maintain infrastructure on an ongoing basis

**1 product manager**
- Determine which new features & functionality to build based on your business’s needs
- Set project milestones for the integration

**1 analyst/data scientist**
- Analyze and determine the optimal rules & thresholds for best ads performance
Gaming automation use cases

Automate insights data pulls

Use the insights API to periodically download the performance data for all campaigns. Automatically pull metrics for all the key metrics that you care about, such as impressions, app installs, CPI, clicks, CTR, etc.

// Download ads performance data
$fields = array(
    'ad_id',
    'ad_name',
    'impressions',
    'ctr',
    'spend',
    'actions',
);

$params = array(
    'date_preset' => 'last_3_days',
    'level' => 'ad',
    'limit' => 5
);

$time_start = microtime(true);

$insights = $account->getInsights($fields, $params);

// Let Cursor fetch next page automatically in the loop
$insights->setUseImplicitFetch(true);

// Store insights data
foreach ($insights as $insight) {
    $data = $insight->getData();

    if ($data['ad_id']) {
        $actions = $data['actions'];
        foreach ($actions as $action) {
            // Process action data
        }
    }
}
if ($action["action_type"] == "link_click") {
    $clicks = $action["value"];
} else if ($action["action_type"] == 'mobile_app_install') {
    $mobile_installs = $action["value"];
}
$cpi = $mobile_installs > 0 ? $spend / (float) $mobile_installs : 0;
$insight = new AdInsight(
    $retrieve_time=$now,
    $ad_name=$data["ad_name"],
    $ad_id=$data["ad_id"],
    $past_3d_CTR=$data["ctr"],
    $past_3d_cost=$data["spend"],
    $past_3d_install=$mobile_installs,
    $past_3d_CPI=$cpi
);
// Store this ad insight into your database
$insight->store();
}

After downloading the insights data, you can store this information in a local database, based on the ad id, for optimization usage.

Configure rules to optimize ads on key metrics

After retrieving the key campaign performance metrics, you can immediately start to analyze our campaign performance and evaluate the results. Using the insights you pulled, you can decide how to manage your ads.

Metrics you can evaluate include:

- CTR
- CPI
- Impressions (to see if the ad is having delivery over time)
- Frequency
- Relevance score
- Spend (monitoring ads based on actual spend instead of the daily budget)

And some of the actions you can take based on these metrics:

- Pause the ad
- Restart the ad (may be previously paused due to daily spend)
- Reallocate budget across adsets
For example, if you want to pause the ads that have past 3 day average CTR lower than 1% or average CPI larger than $10 (depending on the currency setting for the account this number would be in different currencies), you can do the following:

```php
ctr_requirement = 1;
$cpi_requirement = 10;

// Act based on performance data
$insights = getAdInsights();
foreach ($insights as $insight) {
    if ($insight->past_3d_CTR < $ctr_requirement ||
        $insight->past_3d_CPI > $cpi_requirement) {
        $ad_id = $insight->ad_id;
        echo '\nStopping ad: ' . $ad_id;
        $ad = new Ad($ad_id);
        $ad->update(array(
            Ad::STATUS_PARAM_NAME => Ad::STATUS_PAUSED
        ));
    }
}
```

Implement precise targeting

Facebook Custom Audiences are one of the best tools for precise targeting. In acquisition campaigns or re-engagement campaigns, you can utilize Custom Audiences to find out the specific user group so you can show them the most relevant creative. Or, you can use your most valuable users as the seed custom audience to create a group of lookalike audiences and bid your true value for them to maximize ROI.

There are several ways to create Custom Audiences from your game or user data using the Marketing API:

- Create custom audience by filtering App Events
- Create custom audience by directly uploading a user list to the API

```
/**
 * Create MACA: Installers
 */
use FacebookAds\Object\CustomAudience;
use FacebookAds\Object\Fields\CustomAudienceFields;

$rule = array(
    "_application" => $game_id,
    "_eventName" => "fb_mobile_activate_app"
);

$maca = new CustomAudience(null, $account_id);
$maca->setData(array(
```
```php
CustomAudienceFields::NAME => 'Installers',
CustomAudienceFields::SUBTYPE => 'APP',
CustomAudienceFields::RETENTION_DAYS => '180',
CustomAudienceFields::RULE => json_encode($rule),
});
$maca->create();
$installers = $maca->id;
echo "Installers CA: $installers" . PHP_EOL;

/**
 * Create MACA: Finished Tutorial
 */
$rule = array(
    "_application" => $game_id,
    "_eventName" => "fb_mobile_tutorial_completion"
);

$maca = new CustomAudience(null, $account_id);
$maca->setData(array(
    CustomAudienceFields::NAME => 'Finished Tutorial',
    CustomAudienceFields::SUBTYPE => 'APP',
    CustomAudienceFields::RETENTION_DAYS => '180',
    CustomAudienceFields::RULE => json_encode($rule),
));
$maca->create();
$tutorial_finishers = $maca->id;
echo "Tutorial Finishers CA: $installers" . PHP_EOL;

/**
 * Create exclusion targeting
 */
use FacebookAds\Object\Fields\TargetingSpecsFields;
$targeting = array(
    // ...
    TargetingSpecsFields::CUSTOM_AUDIENCES => array(
        array(
            "id" => $installers,
            "name" => "Installers"
        ),
    ),
    TargetingSpecsFields::EXCLUDED_CUSTOM_AUDIENCES => array(
        array(
            "id" => $tutorial_finishers,
            "name" => "Finished Tutorial"
        ),
    );
    // ...
);
```

Marketing integration best practice guide
Alternatively, you can create and update custom audiences directly with the user list. Here is an example:

```php
// use the namespace for Custom Audiences and Fields
use FacebookAds\Object\CustomAudience;
use FacebookAds\Object\Fields\CustomAudienceFields;
use FacebookAds\Object\Values\CustomAudienceTypes;
use FacebookAds\Object\Values\CustomAudienceSubtypes;

// Create a custom audience object, setting the parent to be the account id
$audience = new CustomAudience(null, $account_id);
$audience->setData(array(
    CustomAudienceFields::NAME => 'My Custom Audience',
    CustomAudienceFields::DESCRIPTION => 'Lots of people',
    CustomAudienceFields::SUBTYPE =>
        CustomAudienceSubtypes::CUSTOM,
));

// Create the audience
$audience->create();
echo "Audience ID: " . $audience->id."
"

// Assuming you have an array of emails:
// NOTE: The SDK will hash (SHA-2) your data before submitting
// it to Facebook servers
$emails = array(
    'paul@example.com',
    'luca@example.com',
    'bruce@example.com',
    'peihua@example.com',
);

$audience->addUsers($emails, CustomAudienceTypes::EMAIL);
$audience->read(array(CustomAudienceFields::APPROXIMATE_COUNT));
echo "Estimated Size:"
    . $audience->{CustomAudienceFields::APPROXIMATE_COUNT}."\n";
```

Here the ‘user list’ can be a list of user id’s, emails, phone numbers, device advertiser id’s, and so on. Check here for a complete list:

[https://developers.facebook.com/docs/marketing-api/reference/custom-audience/users/](https://developers.facebook.com/docs/marketing-api/reference/custom-audience/users/)

Finally, Custom Audiences can be used to create lookalike audiences for user acquisition campaigns. The latest API enables you to create tiered lookalike audiences ([https://developers.facebook.com/ads/blog/post/2015/11/23/advanced-lookalikes-creation/](https://developers.facebook.com/ads/blog/post/2015/11/23/advanced-lookalikes-creation/)) in a country by specifying starting and ending ratios for each lookalike audiences. Then, you can provide different bids for the separate tiers in your targeting hierarchy.
Here is one way you can create a lookalike audience of similarity from 1% to 3% in the US:

```php
$lookalike = new CustomAudience(null, $accoun_id);
$lookalike->setData(array(
    CustomAudienceFields::NAME => 'My lookalike audience',
    CustomAudienceFields::SUBTYPE => CustomAudienceSubtypes::LOOKALIKE,
    CustomAudienceFields::ORIGIN_AUDIENCE_ID => $seed_id,
    CustomAudienceFields::LOOKALIKE_SPEC => array(
        'type' => 'similarity',
        'country' => 'US',
        'starting_ratio' => 1,
        'ratio' => 3
    ),
));
$lookalike->create();
```

Optimize ad creative

One challenge that advertisers face is figuring out which ad creative works best for their target audience. Using marketing automation, you can easily test different variations and combinations of your creative to see what resonates best with your audience, and combat ad fatigue with automatic creative refresh.

Automating creative permutations

Let’s say that you have a batch of fresh images you’d like to test, and you want to test with different texts and calls to action, against different groups of target audience. With the help from the Marketing API, the process of combining all these different elements can be automated by iterating through all the combinations. Additionally, you can also control the time frame you wanted to test with the different batch of creatives, and use the optimization functionality to find out the best performing creative to best focus your budget.

Here is an example on the Facebook marketing developers sample site demonstrating automatic creative permutations:
[https://www.facebookmarketingdevelopers.com/samples/adcreation](https://www.facebookmarketingdevelopers.com/samples/adcreation)

Dealing with creative fatigue: automatic Creative Refresh

Another main marketing automation use case related to creatives is the creative refresh. All creatives will have a diminishing CTR after running for a period of time, so at a certain point, you will need to consider updating your ads with a new set of creatives for a given target segment to maintain CTR and ad delivery. Once you have the rule in mind (e.g.: CTR < x% after y days), you
can easily implement this rule into your monitoring and optimization system to accomplish the creative refresh use case.

Here is a simple sample to show this idea:

```php
foreach ($insights as $insight) {

    // Filter out not performing ads and create new ads
    if ($insight->past_3d_CTR < $ctr_requirement) {

        $ad_id = $insight->ad_id;
        $ad = (new Ad($ad_id))->read(array(
            AdFields::ADSET_ID
        ));
        // Get the Ad Set id
        $adset_id = $ad->{AdFields::ADSET_ID};
        echo "\nAd Set ID: " . $adset_id;
        $ad->update(array(
            Ad::STATUS_PARAM_NAME => Ad::STATUS_PAUSED
        ));
        echo "\nStopped old ad: " . $ad_id;

        // Upload new image
        $image = new AdImage(null, $account_id);
        $image->{AdImageFields::FILENAME} = $ad_image_file;
        $image->create();
        echo 'Image Hash: '.$image->hash . "\n";

        // Create the new creative
        $creative = new AdCreative(null, $account_id);
        $creative->setData(array(
            AdCreativeFields::NAME => 'New Creative',
            AdCreativeFields::TITLE => 'Play the game!',
            AdCreativeFields::BODY => 'Fresh new creative!',
            AdCreativeFields::IMAGE_HASH => $image->hash,
            AdCreativeFields::OBJECT_URL => 'http://www.example.com/',
        ));
        $creative->create();
        echo 'Creative ID: '.$creative->id . "\n";

        // Create new ad under the same adset to use the new creative
        $ad = new Ad(null, $account_id);
        $ad->setData(array(
        ));
        $ad->create();
        echo 'Ad ID: '.$ad->id . "\n";
    }
}
```
Utilize LTV bidding

Map user segments with Custom Audiences using the Custom Audiences API

The best practice for winning as much ad delivery opportunities as you can while maintaining optimal ROI is to bid your true value for user segments differently, based on their value or LTV in your game.

You can use the general purchase information—-or even other metrics such as social aspects—-to measure how a user can contribute to your game revenue, and calculate the LTWs from there. Then you can establish different user groups with different evaluations.

Then, you can use the user information (e.g. the mobile device advertising ids), to map these segments from your BI system into Facebook custom audiences. Your BI system should be constantly analyzing the user segments and updating them, sending the new users to the segments to the respective custom audiences you have on Facebook.

// use the namespace for Custom Audiences and Fields
use FacebookAds\Object\CustomAudience;
use FacebookAds\Object\Fields\CustomAudienceFields;
use FacebookAds\Object\Values\CustomAudienceTypes;
use FacebookAds\Object\Values\CustomAudienceSubtypes;

// Create a custom audience object, setting the parent to be the account id
$audience = new CustomAudience(null, $account_id);
$audience->setData(array(
    CustomAudienceFields::NAME => 'My Custom Audiece',
    CustomAudienceFields::DESCRIPTION => 'Lots of people',
    CustomAudienceFields::SUBTYPE =>
    CustomAudienceSubtypes::CUSTOM,
));
Create the audience

```php
// Create the audience
$audience->create();
echo "Audience ID: " . $audience->id."\n";

// Array of device IDs in your user segment
$device_ids = array(
    '96bd03b6-defc-4203-83d3-dc1c730801f7',
    // ...
);

$audience->addUsers($device_ids,
    CustomAudienceTypes::MOBILE_ADVERTISER_ID);
$audience->read(array(CustomAudienceFields::APPROXIMATE_COUNT));
echo "Estimated Size:" . $audience->{CustomAudienceFields::APPROXIMATE_COUNT}."\n";
```

Create lookalike audiences

Next, you could take the Custom Audiences you created and create a set of lookalike audiences with different countries and ratio setting. Here is a sample from Facebook marketing developers site featuring lookalike audience creation:

https://www.facebookmarketingdevelopers.com/samples/multiple_lal

Differential bidding based on target segments

Finally, when you create re-engagement ads, you can use the custom audience you created and use the true value of bringing a user back to the game for the bidding. The targeting and bidding for bringing high spenders vs. lower spenders into the game should be different and not overlap.

For user acquisition campaigns, you can then use the lookalike audiences from your high value users and target the most similar lookalike audiences with the highest bidding, then use the next tier of lookalike audiences with lower bidding price, and finally interest and demographics targeting with lower bidding, to achieve the over all optimal ROI.

Here is a sample of creating campaign and use tiered lookalike targeting strategy from Facebook marketing developers site:

https://www.facebookmarketingdevelopers.com/samples/tiered_lookupalike
Working with Mobile Measurement Partners

There are some general misunderstandings around the different install tracking numbers from MMP, Facebook Analytics and Facebook ad reports. Here are some clarifying differences:

Facebook analytics

Facebook Analytics keeps track of all install pings from Facebook SDKs or MMP pings. The installs that appear on Facebook Analytics are the total installs, regardless of channel.

Facebook ad reports

The Facebook ad reports will only show attributed installs or relevant metrics. The default attribution is 28 days and 1 day click, which is also how Facebook charges. In addition, Facebook ad reports by default report on the impression time (install attributed to the date when the user interacted with the ad) instead of conversion time (installs attributed to the date when the user actually installed the app). For more details on attribution and differences between ad reports and MMP reports you can read this: https://developers.facebook.com/docs/app-ads/resources/faqs

MMP reports

Generally, MMPs can help you de-duplicate across multiple marketing channels, and report to you based on last click attribution, so there may be users who interacted with Facebook ad within the attribution window, but installed after clicking an ad from another channel B. The MMP may tell you that the install came from channel B, while Facebook may also attribute this install to your campaign on Facebook. Additionally, since MMPs use last click attribution, there is no view through installs in MMP reports.