# **Ant Technology**

## Mobile Content Delivery Platform User Guide



Document Version: 20231226



## Legal disclaimer

#### Ant Group all rights reserved © 2022.

No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Ant Group.

#### Trademark statement

<sup>6</sup> <sup>蚂蚁集团</sup> ANT GROUP and other trademarks related to Ant Group are owned by Ant Group. The third-party registered trademarks involved in this document are owned by the right holder according to law.

#### Disclaimer

The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Ant Group reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through channels authorized by Ant Group. You must pay attention to the version changes of this document as they occur and download and obtain the latest version of this document from Ant Group's authorized channels. Ant Group does not assume any responsibility for direct or indirect losses caused by improper use of documents.

### **Document conventions**

Style	Description	Example	
A Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	Danger:     Resetting will result in the loss of     user configuration data.	
O Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	Warning:     Restarting will cause business     interruption. About 10 minutes are     required to restart an instance.	
() Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.	
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	? Note: You can use Ctrl + A to select all files.	
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings> Network> Set network type.	
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click <b>OK</b> .	
Courier font	Courier font is used for commands	Run the cd /d C:/window command to enter the Windows system folder.	
Italic	Italic formatting is used for parameters and variables.	bae log listinstanceid Instance_ID	
[] or [a b]	This format is used for an optional value, where only one item can be selected.	ipconfig [-all -t]	
{} or {a b} This format is used for a required value, where only one item can be selected. switch {active stand}			

## Table of Contents

1.About Mobile Content Delivery Platform	06
2.Terminology	08
3.Quick start	11
4.Integrate MCDP into Android	17
4.1. Quick start	17
4.2. Configure startup page	18
4.3. Configure preset booth	19
4.4. API reference	20
4.5. Advanced operations	25
4.6. Tutorial	26
4.6.1. Tutorial overview	26
4.6.2. Create a native project	27
4.6.3. Create an application in the mPaaS console	34
4.6.4. Access a project in native AAR mode	35
4.6.5. Initialize mPaaS configuration	36
4.6.6. Configure app startup page	37
4.6.7. Configure a preset center booth	38
4.6.8. Verify booth and delivery	39
5.Integrate MCDP into iOS	41
5.1. Add SDK	41
5.2. Use SDK	41
5.3. Configure startup page	43
5.4. Configure preset booth	46
5.5. API reference	48
6.Integrate MCDP into HTML5	52
6.1. Configure preset booth	52

7 Canada anarationa	
7.Console operations	57
7.1. Manage group	57
7.1.1. About user group	57
7.1.2. Create a group	57
7.1.3. Manage groups	58
7.2. Booth management	59
7.2.1. About booth	59
7.2.2. Create a booth	67
7.2.3. Manage booths	70
7.3. Material material	72
7.3.1. Create material	72
7.3.2. Manage materials	73
7.4. Manage activaty	74
7.4.1. About activity unit	74
7.4.2. Create an activity unit	75
7.4.3. Create an activity	75
7.4.4. Manage activities	84
7.4.5. Manage activity units	87
7.5. Push management	89
8.FAQ	91

## **1.About Mobile Content Delivery Platform**

Mobile Content Delivery Platform (MCDP) provides personalized advertising capabilities within the app. It supports personalized advertising for target group and helps app operators accurately reach users in time, to achieve the objectives of customer retention, service promotion, and business growth. In addition, MCDP allows you to apply multiple materials in each advertisement delivery. The system performs automatic optimization based on the advertising effect, to provide you with the optimal advertising traffic. No manual intervention is required, reducing the operating costs and effectively improving the marketing effect.

#### **Benefits**

#### • Saying goodbye to marketing scheduling

MCDP is an all-in-one intelligent marketing management platform. It supports the one-stop configuration of activity group, marketing resources, and marketing actions, and provides marketing data dashboards. app operators can complete the marketing process independently, without relying on R&D personnel, to independently create and execute activity plans. And the operation is simple and fast, therefore bidding farewell to the long marketing scheduling.

#### • Flexible customization of the audience

MCDP integrates the user grouping capabilities of the mobile analysis component (MAS), which can lock on the target group of the activity based on the attributes or behavior of the client users. It also supports the manual creation of the group.

#### • Supporting high concurrent calls

MCDP supports 1,000 concurrent calls per second. The server has 4 CPUs and 8 GB memory) to ensure a stable advertisement delivery effect.

#### Features



The MCDP system framework is shown in the preceding figure. As an all-in-one intelligent marketing management platform, MCDP integrates the creation, execution, management, and feedback of activity. The product functions are as follows:

- Activity management
  - Provides activity units two-level management of activities, to support the launch of more flexible and complex marketing activities.
  - Supports the creation of two types of marketing activities: proactive marketing and interactive marketing.



- Supports adding two kinds of marketing actions, including advertisement delivery and message push, in one activity at the same time.
- Supports completing the delivery configuration of multiple advertisements in one process, which significantly improves the efficiency of activity configuration.
- The fatigue strategy that supports delivery and push can protect the user experience, and also provides a foundation for the parallel launch of multiple activities.
- Supports custom delivery rules to achieve flexible phased delivery.
- Group management

You can create target group for activity advertising, or directly call the group of the mobile analysis component, and deliver ads to group filtered based on user behavior or attributes.

• Booth management

You can create and manage a booth in a visualized manner and create booths with different styles, positions, heights, and content types without client code, reducing development time and costs. MCDP provides the booth data dashboard to monitor the use of booth resources in real time, so that you can adjust the allocation of the resources.

• Material management

Provides a online material library and supports a variety of material types. The materials of advertising needed in marketing activities can be obtained directly by calling resources in the material library, which improves the configuration efficiency of the continuous activity creation process. Labor efficiency is effectively improved through team collaboration.

• Push management

Push management is used to configure push fatigue.

• Activity effect analysis

Provides T+1 activity marketing data, and evaluates the activity effects from multidimensions, helping you learn the marketing situation at a glance.

#### **Application scenarios**

• Proactive marketing

A bank launches a marketing activity for promotion and user acquisition. The theme of the activity is a discount on each online payment on Wednesday. And the advertisement undergoes a one-year recurring delivery every Wednesday. Different groups, different materials delivered.

- For the users who have opened an account and have activated the online payment function: prepare activity A, that is, providing users with an X discount on each online payment. The activity content are delivered to users through MCDP advertising (banner advertisements in the middle of the app home page).
- For the users who have opened an account and have not activated the online payment function: prepare activity B, that is, providing users with three coupons for online payment activation, and offering an X discount on each online payment. The activity content are delivered to users through MCDP advertising (banner advertisements and splash advertisements in the middle of the app home page).
- Interactive marketing

For users who have logged in the app in the last month, if the users log in for 3 consecutive days during the activity period, the activity message "Point rewards for active users" will be pushed to them.

## 2.Terminology

The following terms are sorted in alphabetical order.

#### Α

#### Activity

Also known as the marketing activity, it refers to the time and frequency of material display in the booth, as well as the rules of advertisement delivery, such as the control process of delivery location, platform, application version, and target group. **Activity unit** 

Activity unit refer to a collection of a series of marketing activities created for the marketing purpose. In one activity unit, there can be multiple activities with the same theme. **Advertisement space** 

#### Advertisement space

The position used to display advertising materials in the booth. One advertisement space can display one material, and one booth can contain multiple advertisement spaces.

В

#### Booth

The area on the client page for displaying content (advertising materials).

С

#### **Critical event occurrences**

The number of occurrences of the MAS event associated with the activity after users click the activity advertisements during a certain period of time.

#### D

#### **Deduplicated reach devices**

The sum of the total amount of deduplicated devices that display any advertising materials in the marketing activity and the number of devices that successfully receives pushes during a certain period of time.

#### **Deduplicated share devices**

The total number of unique devices on which the Share button is clicked on the activity landing page during a certain period of time. And the implementation depends on the accuracy of the customer-side tracking.

#### **Device ID**

Advertise to the delineated device IDs (utdid). Regardless of whether the user ID is logged on or not, the delivered advertisements will be displayed, as long as the app is opened on the delineated device.

#### Devices with critical event occurrence

The total number of unique devices on which the MAS event associated with the activity occurs after users click the activity advertisement during a certain period of time. **Display position** 

The position of the booth on the page, such as top and bottom. **Display style** 

The display style of the booth on the page, such as banner and bulletin. **Dynamic booth** 

A booth that is integrated through dynamic configuration. The display process for such a booth must be controlled through the information issued by the server (console).

#### **Event conversion rate**

Refers to the proportion of devices on which the MAS event associated with the campaign



occurs in responding devices.

F

#### Fatigue

Refers to the setting of an upper limit for the total reach of a single user/device in an unit of time.

#### Interactive marketing activity

Targeted marketing and promotion activities triggered by one or more behavior of users in the future. For example, if active users browse wealth management products for 3 consecutive days during the activity period, targeted advertisements for wealth management products will be delivered on the application home page.

#### Μ

#### Multi-frame booth

Refers to the booth with more than 1 advertisement space frames.

Ν

#### Number of advertisement space frames

Refers to the number of advertisement spaces in a booth, that is, the number of campaign materials that can be displayed in the booth when the booth is occupied by the campaign.

#### **Preset booth**

Booths integrated through client configuration. The display positions of such booths must be fixed by codes on the client.

#### Proactive marketing activity

Direct marketing and promotion activities for targeted group. For example, send a push message offering rewards for logon to users who have not logged on in the last 15 days.

#### R

#### **Reach count**

The sum of the number of exposures of an activity advertisement and the number of successful pushes during a certain period of time.

#### **Response count**

The sum of the number of advertisement clicks and the number of push clicks during a certain period of time.

#### Response rate

Refers to the proportion of responding devices in reached devices. **Responding devices** 

The sum of the number of users who click advertisements and the total number of unique devices on which the pushes are clicked during a certain period of time. **Retention rate** 

Retention rate is a statistical indicator used to reflect the operation of websites, Internet applications, or online games. It is the average number of daily active users in the statistical cycle (week/month) who still launch the App on the Nth day. And N is usually 2, 3, 7, 14, and 30, which correspond to the next-day retention rate, three-day retention rate, weekly retention rate, half-month retention rate, and monthly retention rate, respectively.

#### S

#### Share rate

Refers to the proportion of devices with sharing operations in responding devices. **Share count** 

<sup>&</sup>gt; Document Version: 20231226

The number of clicks to share the activity on the landing page during a certain period of time. And the implementation depends on the accuracy of the customer-side tracking. **Single-frame booth** 

Refers to the booth with one advertisement space frames.

#### U

#### User ID

Advertise to the delineated user IDs (userId). When a user logs in to a device by using a grouped user ID, the delivered ad is displayed to the user. When a user logs in to multiple devices by using the same user ID, the delivered ad is displayed on all the devices.

## **3.Quick start**

This topic describes how to quickly complete the entire process from integrating the Mobile Content Delivery Platform (MCDP) component to the client to creating and launching a MCDP marketing activity.

The complete MCDP workflow is as follows:



This topic describes the creation of an interactive marketing activity as an example. Meanwhile, this topic explains some concepts and allocation rules of booth resources involved in marketing activities.

#### Concepts

Before you start integration and using MCDP, you need to understand some basic concepts about MCDP:

- Activity unit: Refers to a collection of marketing activities created around a marketing purpose. In one activity unit, there can be multiple activities with the same theme.
- **Activity**: Refers to marketing activity, which means a process of controlling the duration and frequency for displaying materials in a booth and the advertising rules, for example, the location, platform, application version, and target groups.
- User group: Refers to the target audience of marketing activities in the MCDP.
- **Booth**: Refers to an area on the client page used to display delivered content (advertising materials). You can use one booth for various activities. There are two types of booths, including dynamic booth and client preset booth.
  - Dynamic booth: You can create a booth and complete configurations for all attributes of the booth in the console.
  - Client preset booth: You need to preset configurations for some attributes of the booth by editing codes in the client project, and then create the booth in the console. Enter the booth ID preset in the client project and complete configurations for other attributes of the booth.
- Advertising space: Refers to the space where an advertising material is displayed in the booth. A booth includes multiple advertising spaces, but only one material can be displayed in one advertising space.
- **Material**: Refers to the content displayed in the advertisement, which can be a static image, GIF, or animation.
- **Freshness**: The freshness is measured by time. The closer to the current date, the higher the freshness. For activities, the later the online time, the higher the freshness. For materials, the later the upload time, the higher the freshness.
- **Fatigue**: Refers to the configured upper limit of the total number of times that an activity reaches a single user or device in unit time.

#### Integrate MCDP to client

The integration operation of MCDP depends on the client. Refer to the corresponding description about integration to the client for more details.

#### Android

The operation procedures are as follows (click the text for each step to check the corresponding description):

- 1. Complete integration preparations
- 2. Create an mPaaS application in the console
- 3. Add the configuration file to the project
- 4. Select the integration method and baseline
- 5. Integrate MCDP into project

#### iOS

The operation procedures are as follows:

- 1. Create an mPaaS application in the console
- 2. Select the integration method and baseline
- 3. Integrate MCDP into project

#### **HTML5** pages

See Integrate MCDP into HTML5 to complete MCDP integration.

#### **Prepare activity resources**

#### Create a user group

Log in the mPaaS console and select the created application. Select **Mobile Content Delivery Platform > Group management** in the left navigation bar and click **Create group**. Select the type of group, add relevant users, and click Save.

For more information, see the description document Create a group.

#### Create a booth

Next, you can create a pop-up booth through dynamic configuration.

- 1. In the left navigation bar, select **Mobile Content Delivery Platform > Resource management > Booth management**, and then click **Create a booth**.
- 2. Configure the basic information of the booth, including the booth name, custom ID, the information of page address, and valid (buffer) time. For the description of each parameter, see Booth basic information.
- 3. After you completed the configuration for basic information, you can configure the details for the booth. For the description of each parameter, see Booth configuration information.
- 4. After you confirm all the information, you can click **Submit** to complete the booth creation.

For the creation method of preset booths and splash booths on the startup page, see Create a booth.

#### Create a material

You can create materials in advance and directly use the resources in the material library as advertising materials to simplify the process of creating marketing activities in the future.

In the left navigation bar, select **Mobile Content Delivery Platform** > **Resource management** > **Material management** page and click **Create material**. Configure the material information, including the name, type, and the booth type to which the material will be applied, upload the material, and click Save.

For more information, see Create material.

#### **Create an event**

The interactive marketing activity refers to the targeted marketing activity triggered when a user performs one or more actions. Therefore, before you configure an interactive marketing activity, you must create an event to record the trigger action in Mobile Analysis Service (MAS). See Configure an event for more details. You can skip the step of configuring marketing events for active marketing activities.

You must select **mPaaS console** > **Mobile Analysis Service** > **Custom analysis** > **Custom configuration** to create a MAS event (see Configure an event for operation details). And then you can integrate the MAS component to the client (see Integrate MAS into Android or Integrate MAS into iOS for operation details) and complete the custom event tracking (see Event tracking on the Android client or Event tracking on the iOS client for operation details).

#### Create an activity unit

Create an activity unit to manage marketing activities.

In the left navigation bar, select **Mobile Content Delivery Platform** > **Activity management** and click **Create an activity unit**. Enter the unit name and select the conversion metrics. You can set the conversion metrics as a Mobile Analysis Service (MAS) event related to the activity. The occurrence of the event means that you complete the activity conversion target.

For more information, see Create an activity unit.

#### Create a marketing activity

Create an interactive marketing activity to deliver advertisements or push messages to the target audience.

After you complete the creation of an activity unit, a dialog box of **Select the activity type** will pop up on the page, which means that you are in the activity creation procedure. Therefore, you can directly create an activity.

The operation procedures are as follows:

- 1. Configure the basic information of the activity.
- 2. Configure target groups for the activity. You can select the target group, the group in the allowlist, or the default full group. Meanwhile, in the selected group, you can choose the target group which conforms to conditions including the platform, city, and version.
- 3. Set a marketing activity that triggers a marketing promotion when a user performs this event. As the following configuration, from September 8 of 2020 to October 8 of 2020, if a user taps the wealth management entry once, the system will trigger the action to deliver advertisements or push messages.
- 4. Set marketing actions, including booth delivery and message push.
  - Booth delivery: Select the preset booth and materials. You can select the materials in the material library, or directly upload the local materials. Meanwhile, you can set the fatigue controlling strategy and the booth display priority for the activity. For the allocation of booth resources under the priority strategy, see the following booth resource allocation rules.
  - Message push: Set the title, text, and validity period of the message you need to push.

#### **Booth resource allocation rules**

If you use materials for multiple activities in the same booth, there will be a competition for advertising space resources in the booth.

Allocation rules of advertising space resources in the same booth. The rules described according to priority are as follows:

- Activity priority: Depends on the booth display priority, which is the priority of booth usage for the activity configured manually through the console. That is, activities with higher priority are displayed first.
- Activity online time: When activities with the same priority compete for the same booth, the sorting logic is the activity freshness. That is, the latest online activities are displayed first.
- Additional rules: The activities on the last day of their validity are displayed first. When multiple activities have the same priority under the preceding two rules, the activity materials on the last day of their validity are displayed first. Activities that meet this condition do not need to follow the rule of "activity online time".

Descriptions about the display of activity materials:

- When the number of advertising spaces in the booth is greater than or equal to the number of materials to be displayed in the booth, all materials are sorted and displayed by combining activity priority, activity freshness, and material freshness. Take confirming the display order of advertising materials in a banner advertising space as an example.
- When the number of advertising spaces in the booth is less than the number of materials to be displayed in the booth, all materials are sorted by combining activity priority, activity freshness, and material freshness. And the first N materials of which the number is equal to the number of advertising spaces in the current booth are sorted and displaced.

#### Launch a marketing activity

A created marketing activity does not take effect automatically. You need to manually launch the marketing activity to activate it.

After the activity takes effect, if a specified user in the previous steps taps the wealth management entry in the application, the application will display the corresponding advertisements and push a marketing message to this user when the user enters the application again.

Perform the following steps:

- 1. In the activity unit list on the **Activity management** page, click the name of the activity unit to enter the activity configuration console of the activity unit.
- 2. Click **Activate** in the target activity card to activate it. The status before and after the launching of the activity is as follows:

#### View the activity effect

If marketing data exists during the activity delivery period after an activity goes online, MCDP will perform a statistical analysis of marketing data based on the reported behavior tracking data. MCDP will also generate a marketing data chart of T + 1 days. In the **Operations** column of the activity unit list, you can click **Effect** and select to view the marketing data of all activities or one activity in this unit.

When you select to view one activity, you can also view the activity effect of all marketing channels or one marketing channel under the activity.

Descriptions about each activity data metric are as follows:

Metric

Description

Activity retention	Refers to the retention status of the number of exposed devices for each day in the future 7 days, 15 days, and 30 days.
Activity share count	The number of times that the activity is clicked and shared on the activity landing page within a certain period of time. The specific implementation depends on the accuracy of the event tracking on the client, which requires the activity page to support sharing and have the corresponding event tracking.
Critical event occurrences	The number of MAS events associated with the activity that occur after users clicked the activity advertisements within a certain period of time.
Deduplicated response devices	The sum of the number of users who click advertisements and the total number of deduplicated devices on which pushes are clicked within a certain period of time.
Deduplicated reach devices	The sum of the total number of deduplicated devices displaying any advertising material in a marketing activity and the number of devices successfully receiving pushes within a certain period of time.
Deduplicated share devices	The total number of unique devices on which the Share button is tapped on the activity landing page within a certain period of time. The specific implementation depends on the accuracy of the event tracking on the client, which requires the activity page to support sharing and have the corresponding event tracking.
Devices with critical event occurrence	The total number of deduplicated devices on which the MAS events associated with the activity occur after users clicked the activity advertisements within a certain period of time.
Event conversion rate	The ratio of the number of devices on which the MAS events associated with the activity occur to the number of responding devices.
Reach count	The sum of the number of exposures of activity advertisements and the number of successful pushes within a certain period of time.
Response count	The sum of the number of the clicks of advertisement push within a certain period of time.
Response rate	The ratio of the number of responding devices to the number of reached devices.



Share rate	The ratio of the number of devices with sharing operations to the number of reached devices.

## 4.Integrate MCDP into Android

### 4.1. Quick start

#### Important

mPaaS discontinued maintenance of the 10.1.32 baseline, effective from June 28, 2020. Use 10.1.68 or 10.1.60 series baseline. For how to upgrade the baseline version, see mPaaS 10.1.68 upgrade guide or mPaaS 10.1.60 upgrade guide.

 $\mathsf{MCDP}$  supports two integration modes: native AAR mode and componentized (Portal & Bundle) mode.

#### **Prerequisites**

- The version of the baseline you are using is 10.1.60-beta or higher. mPaaS fully supports MCDP from version 10.1.60-beta of the baseline. Before adding the SDK, make sure that the version of the baseline you are using is 10.1.60-beta or higher. The version information of the baseline can be obtained from the Current mPaaS SDK version information in the mPaaS > Baseline upgrade menu.
- Before you integrate the component to in the native AAR mode, add mPaaS to your project first.
- If you want to connect the component to the mPaaS based on components, you need to first complete the Componentized integration procedure.

#### Add SDK

#### **Native AAR mode**

You can use the **Component management (AAR)** feature to install the **Mobile Content Delivery Platform (MCDP)** component in your project. For more information, see ARR component management.

#### Componentized (Portal & Bundle) mode

In your Portal and Bundle projects, install the **Mobile Content Delivery Platform (MCDP)** component through the **component management** feature.

For more information, see Manage component dependencies.

#### Initialize mPaaS

In the **native AAR** integration mode, you must initialize the mPaaS.

```
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        // mPaaS initialization
        MP.init(this);
    }
}
```

For more details, see Initialize mPaaS.

#### Add configuration

Since the version 10.1.68.18, MCDP does not initialize booths and activities by default when the process starts. You need to manually add the following configuration to the AndroidManifest file to enable automatic initialization:

```
<meta-data android:name="com.mpaas.cdp.autoInit" android:value="true" />
```

After upgrading from a lower baseline version to version 10.1.68.18, you need to manually add configurations to the AndroidManifest file to ensure the normal operation of the corresponding functions.

#### **Follow-up steps**

After the SDK is added, you can preset booths or dynamically configure booths based on business needs.

### 4.2. Configure startup page

Startup-page advertisement is also known as splash advertisement. The startup page, also called splash screen, is displayed after the application is started and the framework initialization is completed and disappeared when the application home page appears.

After configuring the startup page on the client, you can configure Splash booth information and advertising content on the console. For details, see Create a booth and Creating a marketing activity. It's better to configure the fatigue control of the splash booth of the startup page as "disappeared after xx seconds". The application obtains and displays the booth delivery data based on the configuration, and closes the page after xx seconds of the countdown. These steps realize the dynamic distribution and display of the delivery data.

#### ? Note

Due to the asynchronous process of downloading delivery data, only the download operation is performed for the first time after configuring the launch of the startup page in order not to block the launch of the application, the images are cached locally, and the cached images will be displayed the next time the application is started.

Based on the mPaaS framework, the timing and rationale of the startup page are as follows. For details about the mPaaS framework, see mPaaS framework introduction.

- 1. After the framework started, the main thread creates and initializes LauncherActivityAgent , and opens the home page through the callback of LauncherActivityAgent.postInit .
- 2. Check on the home page and open the startup page.

#### Examples

1. Initialize the startup page in the home page.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Home page logic
    // ......
    // ......
    // ......
    // ......
    if (SplashActivity.checkIfSplashPrepared()) {
        startSplash();
    }
}
private void startSplash() {
    startActivity(new Intent(this, SplashActivity.class));
    overridePendingTransition(0, 0); // Remove the transition animation
}
```

2. Display the splash screen on the SplashActivity of the startup page.

```
private void doSplash() {
         final CdpAdvertisementService cdpAdvertisementService = cpdService();
         cdpAdvertisementService.doSplash(this, new HashMap<String, String>(), new Cd
pAdvertisementService.IAdEventHandler() {
             Override
             public void onClosed(SpaceInfo spaceInfo) {
             }
             ROverride
             public void onJump(SpaceInfo spaceInfo) {
                 // Redirect to the active target page
             }
         });
     }
     public static CdpAdvertisementService cpdService() {
        CdpAdvertisementService serviceByInterface =
LauncherApplicationAgent.getInstance().getMicroApplicationContext().findServiceByInterf
e (
             CdpAdvertisementService.class.getName());
         return serviceByInterface;
     }
```

## 4.3. Configure preset booth

Configure preset booth means to control the display of the booth (for example, the page where the booth is located and the location of the booth) through the client-side code, and at the same time obtaining other booth information from the server through calling API, such as booth height, display style, content type, etc.

#### About this task

You can preset the CdpAdvertisementView booth parameters through XML to the layout.xml file on the interface where the booth is located in the App, or add the parameters to the layout.xml file with Java code. Then, calling the updated interface during the life cycle of onResume () in Activity.

⑦ Note
The booth information obtained from the server will be filled in the CdpAdvertisementView parameter.

If you are a new user of the Mobile Content Delivery Platform, we recommend that you dynamically configure the Android page of the advertising booth directly on the console (server). For more server configuration information, please refer to Create a booth. **Procedures** 

1. Preset CdpAdvertisementView in the layout by one of the following methods.

```
• Preset CdpAdvertisementView parameters via XML:
```

```
<com.mpaas.cdp.CdpAdvertisementView
android:id="@+id/pit_cdp_cav"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
```

• Add the CdpAdvertisementView parameter with Java code:

```
CdpAdvertisementView cdpView = new CdpAdvertisementView(context);
parentView.addView(cdpView);
```

2. Call the update interface.

```
cdpView.updateSpaceCode("my_space_code");
```

## **4.4. API reference**

This topic explains the APIs that involved in the access of Mobile Content Delivery Platform (MCDP).

- CdpAdvertisementService: The CdpAdvertisementService in the middle layer of mPaaS encapsulates the interfaces of the MCDP. The related MCDP functions can be realized by calling these interfaces.
- ActionExecutor: This interface implements the interception of booth and advertisement information. When intercepted, the client will not display the corresponding booths and advertisement contents; when not intercepted, the booths and advertisement contents will be displayed.
- Refresh: Implement global status update. Used to re-request ads in scenarios where users change regions, user IDs, etc.

#### CdpAdvertisementService

```
/**
 * Cdp ad serving interface
 *
 */
public abstract class CdpAdvertisementService extends ExternalService {
```

/\*\* \* Initialize all advertising information \* @param extInfo extended information \* @param callBack callback \*/ public abstract void initialized(Map<String, String> extInfo, IAdGetSpaceInfoCallBack callBack); /\*\* \* Query the booth through the booth ID, and asynchronous callback onSuccess (Space Info spaceInfo) returns the query result. \* If there is no local cache, the RPC query result will be called back once. \* If there is a local cache and the update time is not up, the local query result will be called back once. \* If there is a local cache and the update time is up, the local query result will be called back once, and the updated result will be called back again after the RPC is completed. \* The onFail() interface will be called back only when the RPC fails. \* \* @param spaceCode booth ID, you need to apply to the platform \* @param callback callback interface \*/ public abstract void getSpaceInfoByCode(String spaceCode, IAdGetSingleSpaceInfoCallBack callback); /\*\* \* Query the booth through the booth ID, and asynchronous callback onSuccess (Space Info spaceInfo) returns the query result. \* If there is no local cache, the RPC query result will be called back once. \* If there is a local cache and the update time has not arrived, the local query r esult will be called back once. \* If there is a local cache and the update time is up, the local query result will be called back once, and the updated result will be called back again after the RPC is completed. \* The onFail() interface will be called back only when the RPC fails. \* @param spaceCode booth ID, you need to apply to the platform \* @param extInfo extended parameters \* @param immediately only returns the RPC result \* @param callback callback interface \*/ public abstract void getSpaceInfoByCode(String spaceCode, Map<String, String> extIn fo, boolean immediately, final IAdGetSingleSpaceInfoCallBack callback); /\*\* \* Batch query booths by booth ID list, asynchronous onSuccess(List<SpaceInfo> adSpaceInfo) returns the query results.  $^{\star}$  If there is no local cache, the RPC query result will be called back once. \* If there is a local cache and it is not yet time to update, the local query resu It will be called back once. \* If there is a local cache and it is time to update, the local query result will be called back once, and the updated result will be called back again after the RPC is completed.

\* The onFail(List<String> adSpaceCodes) interface will be called back only when th e RPC failed. \* @param spaceCodeList booth ID list, you need to apply to the platform \* @param extInfo extended parameters \* @param immediately only returns the RPC result \* @param callback callback interface \*/ public abstract void batchGetSpaceInfoByCode(List<String> spaceCodeList, Map<String, String> extInfo, boolean immediately, final IAdGetSpaceInfoCallBack callbac k); /\*\* \* H5 interface \* Query and display ads, currently only called by AdH5Plugin \* @param activity current page \* @param parentView parent control \* @param url address \* @param h5Param parameter \*/ public abstract void checkAndShowAdInH5(final Activity activity, ViewGroup parentVi ew, String url, String h5Param); /\*\* \* Remove the ad view with specified spaceCode \* @param activity Pages whose announcement need to be removed \* @param spaceCode advertising ID \*/ public abstract void removeAdvertisement(Activity activity, String spaceCode); /\*\* \* Get the action executor, return null if it is not set \* @return action executor \*/ public abstract ActionExecutor getActionExecutor(); /\*\* \* Set the action executor \* \* @param executor action executor \*/ public abstract void setActionExecutor(ActionExecutor executor); /\*\* \* Set user ID \* \* @param userId user ID \*/ public abstract void setUserId(String userId);

99敗集团 ANTGROUP rm

```
/**
    * Get user ID
    *
    * @return user ID
    */
   public abstract String getUserId();
    /**
    * Get the callback class of the advertising booth
    */
   public interface IAdGetSpaceInfoCallBack {
       /**
        * Successfully obtained booth information
        *
        * @param adSpaceInfo booth list
        */
       void onSuccess(List<SpaceInfo> adSpaceInfo);
        /**
        * Failed to obtain booth information
        *
        * @param adSpaceCodes Code list of booth request
        */
       void onFail(List<String> adSpaceCodes);
    }
    /**
    * Get the callback class of a booth
    */
   public interface IAdGetSingleSpaceInfoCallBack {
       /**
        * Successfully obtained booth information
        * @param spaceInfo Booth information
        */
       void onSuccess(SpaceInfo spaceInfo);
       /**
        * failed
        */
       void onFail();
   }
}
```

#### ActionExecutor

Mobile Content Delivery Platfo rm

```
/**
* Action processor
 *
*/
public interface ActionExecutor {
   /**
    * Whether to intercept Action
    * @param spaceInfo Booth information
    * @param spaceObjectInfo Advertising information
    * @param url action url
    * @return Return true for successful execution, false for exception
    */
   boolean interceptAction(final SpaceInfo spaceInfo, final SpaceObjectInfo
spaceObjectInfo, final String url);
   /**
    * Execute Action
    *
    * @param spaceInfo Booth information
    * @param spaceObjectInfo Advertising information
    * @param url action url
    * @return Return 1 for successful execution, other values for exception
    */
   int executeAction(final SpaceInfo spaceInfo, final SpaceObjectInfo spaceObjectInfo,
final String url);
}
```

#### Refresh

```
/**
  * Refresh all status callback
   *
   */
public interface IRefreshZoneCallBack{
     /**
      * Start
       *
      */
     void onStart();
      /**
      * End
      */
     void onEnd();
 }
/**
   * Re-request ads when the user changes region, user id, etc.
  * @since 3.0
  * @param callBack
  */
public abstract void refresh(IRefreshZoneCallBack callBack);
```

### 4.5. Advanced operations

After integrating the MCDP component and completing the basic booth configuration, you can also make some other settings, such as controlling the maximum range of the popup, forcing a complete refresh of all booths, setting expansion parameters for RPC requests. The following shows the details of these advanced operations.

#### Control the maximum range of the popup

- Applicable scenario: When you want a larger pop-up page.
- Applicable SDK version: 10.1.60 and above.
- Operation: Preset via XML.
  - Specify the maximum height of the pop-up page:

```
<meta-data
android:name="com.mpaas.cdp.space.height"
android:value="xxx" />
```

• Specify the maximum width of the pop-up page:

```
<meta-data
android:name="com.mpaas.cdp.space.width"
android:value="xxx" />
```

• Specify whether the resolution of the pop-up page is in pixel or dip. The default is dip:

```
<meta-data
android:name="com.mpaas.cdp.space.dpmode "
android:value="dp" />
```

#### Force a complete refresh of all booths

- Applicable scenario: When the user switches account or region during usage.
- Applicable SDK version: 10.1.68 and above.
- Operation: Call the API.
  - i. Obtain the CDP interface service, so that follow-up related functions can use this service to call.

```
private CdpAdvertisementService findCdpApiService() {
    return LauncherApplicationAgent.getInstance()
        .getMicroApplicationContext()
        .findServiceByInterface(CdpAdvertisementService.class.getName());
}
```

ii. Update and refresh all CDP booth data.

findCdpApiService().refresh(new CdpAdvertisementService.IRefreshZoneCallBack(){});

#### Set extension parameters for RPC requests

Applicable SDK version: 10.1.68 and above.

```
MCdpApi.API.api().setExportApi(new IMCdpExportApi (){
   public Map<String, String> getExtras() {
      HashMap<String, String> extras = new HashMap<>();
      extras.put("cdp_extend_params_xxxx", String.valueOf(1));
      return extras;
   }
});
```

### 4.6. Tutorial

### 4.6.1. Tutorial overview

MCDP is supported in two access modes: native AAR and component-based. For details about the two access modes, see Access method introduction.

These steps use the native AAR mode as an example to demonstrate the complete process from creating an Android native project, connecting mPaaS to the project, connecting the MCDP component for code development to using the MCDP function. In these steps, you will learn how to configure the app startup page and how to preset a booth in the middle through the client.

#### **Tutorial description**

These steps assume that you have completed the following preparations:

• Development environment. In these steps, the Windows development environment is taken as an example to demonstrate.



- Web browser. The Chrome browser is recommended.
- An Android phone running Android 4.3 or later, and the supporting data cable.
- Create an Alibaba Cloud account and launch mPaaS products.

#### **Procedures**

These steps consist of the following seven parts:

- 1. Create a native project
- 2. Create an application on the mPaaS console
- 3. Access a project in the native AAR mode
- 4. Initialize mPaaS configuration
- 5. Configure the app startup page
- 6. Configure a booth in the middle
- 7. Verify booth configuration

### 4.6.2. Create a native project

In this topic, you will create an application on which a toast pops up when you click the text, and get a signed APK package. This process mainly includes the following four steps:

- 1. Create a project
- 2. Compile the project
- 3. Create a signature file and sign the project
- 4. Install the app on a mobile phone

If you already have a native Android project and the project has already been signed, redirect directly to Create an application in the mPaaS console.

#### Prerequisites

Before starting to create a project, make sure you have installed the mPaaS plug-in.

#### **Create a project**

1. Start Android Studio, and click File > New > New Project.

*	<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	Analy <u>z</u> e	<u>R</u> efac	tor	<u>B</u> uild	R <u>u</u> n	<u>T</u> ools	VC <u>S</u>	<u>W</u> indow	<u>H</u> elp	mPaaS
	Ν	lew					•		New Pr	roject.					
5		)pen									nPaaS p	roject		bu	ild.gradle (
age	- 🛃 P	rofile	or deb	oug APK					Import	Proje	ct				



2. In the Create New Project dialog box that appears, select **Empty Activity**, and click **Next**.



3. Set the project name (Name), package name (Package name), and storage path (Save location). These steps take the "mPaaS CDP" project as an example. Select API 18: Android 4.3 (Jelly Bean) as the Minimum SDK.

👅 Create New Project		×
onfigure Your Proje	ct	
	Name	
	mPaaS CDP	
÷	Package name	
	com.example.mpaascdp	
	Save location	
	d:\AndroidStudioProjects\mPaaSCDP	
	Language	
	Java	
	Minimum SDK API 18: Android 4.3 (Jelly Bean)	
Empty Activity	Your app will run on approximately 98.4% of devices. Help me choose	
Creates a new empty activity.	Use legacy android.support libraries 🕜	
The application name for most apps begins with	an uppercase letter	
	Previous Next Cancel	Finish

4. Click **Finish** to complete the project creation.

#### **Compile the project**

1. Open the activity\_main.xml file and refer to the following codes to add a button.

```
<Button

android:id="@+id/button"

android:layout_width="101dp"

android:layout_height="50dp"

android:layout_marginStart="142dp"

android:layout_marginTop="153dp"

android:layout_marginBottom="151dp"

android:text="Button"

app:layout_constraintStart_toStartOf="parent" />
```

993年3 Mobile Content Delivery Platfo rm



2. Open the MainActivity class and add a click event for the button.



3. After a successful compilation, you have finished the code writing.

#### Create a signature file and sign the project

When configuring the component code in the mPaaS console, you need to use the signed APK file.

The steps to sign for the project and generate the APK file are as follows:



1. In Android Studio, choose **Build > Generate Signed Bundle / APK**.



2. In the dialog box that appears, select **APK**, and click **Next**.

🛎 Generate Signed Bundle or APK	×
O Android App Bundle	
Generate a signed app bundle for upload to app stores for the following ben	efits:
<ul> <li>Smaller download size</li> <li>On-demand app features</li> <li>Asset-only modules</li> </ul>	
Learn more	
• АРК	
Build a signed APK that you can deploy to a device	
Previous Next Cancel Hel	p



#### 3. Select Create new.

Ă Generate Signed E	Bundle or APK X
Module	The app rest of the second sec
Key store path	Create new Choose existing
Key store password	
Key alias	
Key password	
	Remember passwords
	Previous Next Cancel Help

4. Enter all the required information, and click **OK** to complete signature creation. You can obtain the generated signature file in the specified **Key store path**.

👅 New Key Store					×
Key store path:	):\code	\cdp_key.jks			<b>-</b>
Password:	•••••		Confirm:	•••••	
Кеу					
Alias:	key0				
Password:	•••••	•	Confirm:	•••••	
Validity (years):	2	25 🜲			
Certificate					
First and Last N	lame:	Zhang San			
Organizational	Unit:	mpaas			
Organization:		mpaas			
City or Locality:	City or Locality:				
State or Province:		Zhejiang			
Country Code (XX):		86			
L				ок	Cancel
				OR	Cancer

5. After the fields are automatically entered, click **Next** to start signing for the project.

🛎 Generate Signed E	Bundle or APK X
Module	app 🔻
Key store path	D:\code\cdp_key.jks
	Create new Choose existing
Key store password	•••••
Key alias	key0 📂
Key password	•••••
	Remember passwords
	Previous Next Cancel Help

Select Build Variants as needed, and check the encrypted version V1 (Jar Signature).
 Note that V1 (Jar Signature) is required while V2 (Full APK Signature) is optional.

👅 Generate Signed	Bundle or APK X
Destination Folder:	\User: \user: \undroidStudioProjects\mPaaSCDP\app
	debug
	release
Build Variants:	
Signature Versions:	V1 (Jar Signature) 🗌 V2 (Full APK Signature) Signature Help
	Previous Finish Cancel Help

7. Click **Finish**. After a while, in the debug folder ( ~\mPaaSCDP\app\debug ) under the project folder, you can get the signed APK package of the project. In these steps, the name of the package is app-debug.apk .

#### Install the app on a mobile phone

1. Connect a mobile phone to the computer and enable the USB Debugging mode of the mobile phone.



2. In Android Studio, run the project to install the application on your phone.



3. Open the app and tap **BUTTON**. The toast "Hello mPaaS!" appears, as shown at the bottom of the screen. The display of this toast indicates that the app is installed and can function as expected. At this point, you have successfully installed the application on your phone.

下午5:16 📓 🖺 🔉 😫 Р 🏹 🤌 Р 🔘 0.1K/s 🕸 🗵 🍣 🎟 ≁
mPaaS CDP
BUTTON
DOTTOR
mPaaS CDP: Hello mPaaS!

# 4.6.3. Create an application in the mPaaS console

After completing the project creation and generating a signed APK file, go to the mPaaS console to create an application.

#### Procedure

1. Log in to mPaaS console.



- 2. Click **Create Application**. There is no limit to the number of applications that you can create on the console. And you can create applications without any cost.
- 3. Complete the application information.
  - i. (Required) Enter the application name. Example of application name: mPaaS MCDP.
  - ii. (Optional) Click + to upload the application icon. If you do not upload the icon, a default icon will be used.

The size of the application icon should be less than 1 MB. The aspect ratio is 1:1. The dimension is between 50 pixels and 200 pixels. And the picture format should be JPG or PNG.

- 4. Click **Create** to complete the application creation. You can see the application list. It includes the applications you created.
- 5. Enter the newly created application, choose Code Configuration > Android > Download the Initialize Configuration Now. Enter the Package Name ( com.example.mpaascdp is taken as an example here). And upload the compiled and signed APK package.
- 6. After the configuration completed, click **Download Configuration** at the bottom right of the code configuration panel to download the mPaaS configuration file. The downloaded configuration file is a compressed package. After unzipping this package, you will get a configuration file in .config format and an encrypted picture in .jpg format.

#### **Follow-up steps**

Access a project in the native AAR mode

### 4.6.4. Access a project in native AAR mode

This topic describes how to connect mPaaS to a project in the native AAR mode.

#### Procedure

- 1. In Android Studio, open the existing project, and choose **mPaaS** > **Native AAR mode**.
- 2. In the pop-up access panel, click **Start Importing** under the **Import App Configuration**.
- 3. In the **Import mPaaS Configuration File** dialog box that appears, select **I have** downloaded the configuration file on the console and get ready to import the file to the project, and click Next.
- 4. Select the configuration file downloaded after you create the mPaaS application on the console, and click **Finish**.
- After the process finished, you will receive a message indicating that the configuration file has been imported successfully. At this point, you have completed the process of importing the configuration file manually.
- 6. Click **Start Configuration** under **Access/Upgrade the baseline** on the right side of the window.
- 7. In the **Select an mPaaS baseline version** dialog box that appears, select 10.1.68 from the drop-down list, and click **OK** to access the mPaaS SDK.

? Note

You can click Start Configuration again to upgrade the baseline.

8. Click **Start Configuration** under **Configure/update components** on the right side of the window.

<sup>&</sup>gt; Document Version: 20231226

9. In the component list that appears, check **MCDP**, and click **OK** to add the MCDP component to the project.

At this point, you have successfully accessed the MCDP SDK into the project in the native AAR mode.

#### Follow-up steps

Initialize configuration

### 4.6.5. Initialize mPaaS configuration

These steps use the native AAR mode to access the MCDP component. After adding the MCDP SDK, you need to initialize the mPaaS.

#### Initialize mPaaS

- 1. Create a new MyApplication class in the project.
- 2. In the MyApplication class, add the following codes.

```
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        // mPaaS initialization
        MP.init(this);
    }
}
```

For more details, see Initialize mPaaS.

#### Add Manifest configuration

#### ? Note

If your baseline version is less than 10.1.68.18, this step can be ignored.

By default, from version 10.1.68.18, the MCDP component will not initialize the booth pulling and activities when the process starts. You must manually add the following configuration to the AndroidManifest file to enable automatic initialization:

<meta-data android:name="com.mpaas.cdp.autoInit" android:value="true" />

After upgrading from a lower baseline version to version 10.1.68.18, you must manually add configurations to the AndroidManifest file to ensure the normal operation of the corresponding function.

At this point, you have completed the initial configuration of MCDP. And you can customize the dynamic booth and the visual configuration of marketing activities through the console later.

#### ? Note

Before configuring the booth, make sure you have completed the development of the App page where the booth is located.
#### Integrate related component SDK

In order to provide in-app personalized and targeted advertising capabilities, and help users achieve differentiated precision marketing, MCDP integrates the user grouping capability of the Mobile Analysis Service (MAS) and the push capability of the Message Push Service (MPS). Therefore, before using the MCDP function, we recommend that you access MAS and MPS.

- For details about how to add MAS, see Access Mobile Analysis Service.
- For details about how to add MPS, see Access Message Push Service.

#### **Follow-up steps**

Configure preset booths or configure dynamic booths based on business needs.

# 4.6.6. Configure app startup page

To deliver advertisements on the startup page of the app, namely the splash advertisements, you need to configure the startup page on the client.

The startup page is displayed after the application is started and the framework initialization is completed, and disappears when the application home page appears. The splash booth of the startup page is a special kind of preset booth. You need to configure the startup page through the client in advance, and then configure the advertising content of the splash booth of the page on the console.

How to configure the startup page is as follows.

#### **Procedures**

- 1. Create an XML page for the startup page in the project, and create the SplashActivity class corresponding to the page at the same time.
- 2. Initialize the startup page in the home page. In the MainActivity class, add the following codes for configuration:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   // Home page logic
   // .....
   // .....
   // .....
   if (SplashActivity.checkIfSplashPrepared()) {
      startSplash();
   }
}
private void startSplash() {
   startActivity(new Intent(this, SplashActivity.class));
   overridePendingTransition(0, 0); // Remove the transition animation
}
```

These steps will set a button on the home page of the application. Then you can call the startup page by clicking the button.

3. Display the splash screen in the startup page. In the SplashActivity class, add the following codes for configuration:

```
private void doSplash() {
     final CdpAdvertisementService cdpAdvertisementService = cpdService();
     cdpAdvertisementService.doSplash(this, new HashMap<String, String>(), new CdpAdv
ertisementService.IAdEventHandler() {
         QOverride
         public void onClosed(SpaceInfo spaceInfo) {
        }
         @Override
         public void onJump(SpaceInfo spaceInfo) {
             // Redirect to the active target page
         }
     });
 }
public static CdpAdvertisementService cpdService() {
     CdpAdvertisementService serviceByInterface =
LauncherApplicationAgent.getInstance().getMicroApplicationContext().findServiceByInterf
e(
         CdpAdvertisementService.class.getName());
    return serviceByInterface;
 }
```

4. In the AndroidManifest file, add SplashActivity .

At this point, you have completed the startup page configuration.

#### **Follow-up steps**

After configuring the startup page on the client, you can create a splash booth for the startup page on the console for the purposes of creation and calls. For specific operations, see Create a splash booth.

Note that there can only be one startup-page splash booth in an application.

# 4.6.7. Configure a preset center booth

Special booth positions such as the middle cannot be configured through the mPaaS console. To create such booths, you need preset the booth on the client page through codes to control the booth page, position, and other attributes.

The following describes how to configure a preset booth situated in the middle by client coding.

#### Procedures

- In the project, create an app page that will be used for advertising and its corresponding category, such as AdvertisementBoothActivity. A preset booth will be configured in the middle of the page through the client code.
- 2. Preset the CdpAdvertisementView parameter to the app page created in Step 1 by any of the following methods.

```
• Preset the CdpAdvertisementView parameter through XML:
```



```
<com.mpaas.cdp.CdpAdvertisementView
android:id="@+id/pit_cdp_cav"
android:layout_width="match_parent"
android:layout_height="150px"
android:layout_marginTop="800px"
android:background="#66ccff"
/>
```

• Add the CdpAdvertisementView parameter through Java:

```
CdpAdvertisementView cdpView = new CdpAdvertisementView(context);
parentView.addView(cdpView);
```

3. Call the update API in the onResume () life cycle of the corresponding Activity to obtain other information about the booth from the server, such as booth height, display style, and content type.

cdpView.updateSpaceCode("my\_space\_code"); //Replace my\_space\_code with booth ID

#### ? Note

After completing the corresponding booth configuration on the console, the booth information obtained from the server will be populated into the CdpAdvertisementView parameter.

4. In the AndroidManifest file, add AdvertisementBoothActivity .

At this point, you have completed the configuration of a preset booth.

#### What to do next

After configuring the booth ID, booth page, booth position and other attributes on the client, you can create the corresponding booth on the console (the preset booth ID created on the console must be consistent with that on the client) for use in an activity. For details, see Create a client preset booth.

# 4.6.8. Verify booth and delivery

The configuration of the startup page and the preset booth has been completed on the client. You then need to verify whether the advertisement can be displayed normally in the splash booth and the preset booth of the startup page by creating a marketing activity and placing advertising materials in the middle of the startup page and the corresponding app page.

The entire verification process mainly includes three steps:

- 1. Create a booth
- 2. Create and launch an activity
- 3. Check the effect of mobile advertising

The steps are described in details as follow.

#### Create a booth

- 1. Log in to the mPaaS console and enter the target application mPaaS CDP.
- 2. Create a splash booth for a startup page and a client preset booth by referring to Create a booth.

The splash booth ID of the start page must be **STARTPAGE**, and the ID of the client preset booth must be the same as that configured on the client.

3. Configure booth information and submit the information after completion.

#### ? Note

If the width and height of the booth are set on both the client and the console, the actual display will be subject to the settings on the client.

#### Create and launch an activity

- 1. Create an active marketing activity by referring to Create an activity. When configuring the marketing actions of the campaign, add the two booths created in the preceding steps, and add the corresponding types of materials to the booths. The following shows an example.
- 2. After completing the configuration, click **Submit**.
- 3. On the activity management page, click **Online** on the activity card to launch the activity.

#### Check the effect of mobile advertising

- 1. Install the application mPaaS CDP on the mobile device used for testing.
- 2. After installation, open the application and enter the home page.
- 3. Two buttons are added on the home page in this tutorial. Clicking the buttons will take you to the corresponding startup page and app page.

Normally, the application will display the splash advertisement and the banner advertisement in the middle of the startup page and the specified app page respectively. Otherwise, refer to FAQ for troubleshooting.

# **5.Integrate MCDP into iOS** 5.1. Add SDK

To integrate the MCDP component into iOS client, you need to add the MCDP SDK to the iOS project, and complete the project configurations.

#### ! Important

```
Since June 28, 2020, mPaaS has stopped support for the baseline 10.1.32. Please use 10.1.68 or 10.1.60 instead. For how to upgrade the baseline from version 10.1.32 to 10.1.68 or 10.1.60, see mPaaS 10.1.68 upgrade guide or mPaaS 10.1.60 upgrade guide.
```

#### Prerequisites

You have integrated your project to mPaaS. For more information, refer to Integrate based on native framework and using Cocoapods.

#### Add SDK

Use CocoaPods plugin to add the MCDP SDK. Complete the following steps:

- 1. In the Podfile file, use mPaaS\_pod "mPaaS\_CDP" to add dependency.
- 2. Execute pod install to complete integrating the SDK.

# 5.2. Use SDK

You can refer to the related code snippets to learn the usage of the framework and MCDP.

For the download address and usage method of the code example, refer to Obtain code sample. For API description, refer to API description.

#### Start MCDP

After you start the App, you can call the start method to initialize and start MCDP.

The method is as follows:

```
[CDPPromotionInterface start];
```

#### ? Note

```
We recommend that you add the code to the - (void) application: (UIApplication *) application beforeDidFinishLaunchingWithOptions: (NSDictionary *) launchOptions; method.
```

#### **Configure MCDP SDK**

You can configure the MCDP SDK according to actual needs. For example, you can set the notification upon user's logon and logout.

The method is as follows:



```
// Configure the notification name upon a user's logon. This parameter has no default v
alue.
[CDPPromotionInterface sharedInstance].loginNotificationName = @"xxxx";
// Configure the notification name a user's logout. This parameter has no default value
[CDPPromotionInterface sharedInstance].logoutNotificationName = @"xxxx";
// Configure whether to use location information.
[CDPPromotionInterface sharedInstance].useLbsInfo = YES;
```

## **Register container plugin**

To use relevant JSAPIs on the HTML5 pages, you need to register the container plugin of MCDP.

Content delivery to HTML5 pages depends on HTML5 container (Nebula). HTML5 container initialization is required before you register the container plugin. Just add the following code to - (void) application: (UIApplication \*) application beforeDidFinishLaunchingWithOptions: (NSDictionary\*)launchOptions in DTFrameworkInterface to implement the container initialization. See Initialize HTML5 container to learn more.

```
// Initialize container
[MPNebulaAdapterInterface initNebula];
```

Refer to Custom plug-in and register the container plugin used by MCDP as shown in the following figure.

Root		Dictionary	(3 items)	
▶ JsA	piRuntime	Dictionary	(1 item)	
▼ Plug	ginRuntime	Dictionary	(1 item)	
7	Plugins	Array	(46 items)	-
	▼ Item 0	Dictionary	(3 items)	
	scope	String	scene	
	name	String	PromotionJsPlugin	
	▼ events	Array	(1 item)	
	▼ Item 0	Dictionary	(2 items)	
	name	String		
	useCapture	Boolean	NO	
	▶ Item 1	Dictionary	(3 items)	

Dictionary (3 items)

## **Report data sharing**

This function allows you to report data sharing on the advertisement page. When a user completes sharing, the sharing action and related data is reported by calling the sharing reporting API. After this function is integrated, you can observe the corresponding data indicators on the dashboard of the console.

The procedure is as follows:

1. Listen to the notification of tapping to share event.

```
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(shareAd:)
name:MPCDPDidShareAdNotification object:nil];
```

2. Perform sharing in notification processing, and call the sharing reporting API after sharing is performed.

```
- (void) shareAd: (NSNotification *) notification {
    NSString *adUrl = notification.object;
    // Initiate sharing
    // ...
    // Report advertisement sharing
    [[CDPPromotionInterface sharedInstance] reportShareAdAction];
}
```

# 5.3. Configure startup page

Startup-page advertisement is also known as splash advertisement. The startup page (also referred to the splash) shows after the app starts loading until the system notification of LaunchScreen ends. The startup page disappears when the home page appears.

After you configure the startup page on the client, you can configure the splash booth and advertising content on the console (See Create a booth and Create a marketing activity for details). The application will obtain and display booth delivery data based on the configuration, which realizes data dynamic delivery and display.

#### ? Note

Due to the asynchronous process of downloading delivery data, after configuring the delivery of the startup page, only the download operation is performed for the first time, the images are cached locally, in order not to block the launch of the application. And the cached images will be displayed the next time the application is started.

Based on the mPaaS framework, the timing and principle of the startup page are as follows. For details about the mPaaS framework, see mPaaS framework introduction.

- 1. When the framework starts, the framework will create a bootloader to manage the main window of the application.
- 2. When startup page appears, the framework will automatically switch the window, and return the created startup page window to the user.
- 3. When the bootloader completes the startup and the micro application Launcher completes the display, the framework will close the startup page and switch back to the main window.

#### Prerequisites

Make sure that you have correctly started the CDP component. See Start component for details.

#### Procedures

In the framework classification file DTFrameworkInterface+MPCDPDemo\_plugin.m generated by the mPaaS framework (as shown in the following figure), complete the following configuration.



1. Declare a static variable to hold the window object of the startup page.

static UIWindow \*splashScreenWindow;

2. In the application:handleDidFinishLaunchingWithOptions: method of the file, implement the startup page advertisement logic and open the startup page.

```
- (DTFrameworkCallbackResult) application: (UIApplication *) application
handleDidFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
        // Advertisement logic.
      // Check if the startup page exists.
        BOOL showSplashWindow = YES;
      showSplashWindow = splashScreenExist(showSplashWindow);
        if (showSplashWindow) {
             weak typeof(self) weakSelf = self;
        // Open the startup page.
            splashScreenWindow = APSplashScreenStart(^{
            // The callback for closing the startup page.
                [weakSelf splashScreenDidDismiss];
            });
        }
        return DTFrameworkCallbackResultContinue;
    }
```

3. Implement the logic of startup page closing.

This includes switching the main window of the application, releasing the startup page window, and sending a notification about closing the startup page. Note that the last operation is optional.

```
- (void)splashScreenDidDismiss {
        // Restore the main window of the application as the key window.
        [DTContextGet().window makeKeyAndVisible];
        [self performSelector:@selector(doDismiss) withObject:nil afterDelay:0.0];
    }
    - (void)doDismiss {
        // Release the startup page object.
        splashScreenWindow.rootViewController = nil;
        splashScreenWindow = nil;
        [self notifySplashScreenDismiss];
    }
    - (void)notifySplashScreenDismiss {
        // Send a notification about splash screen closing. This operation is
optional.
        [[NSNotificationCenter defaultCenter]
postNotificationName:@"kSplashScreenDidDismiss" object:nil];
    }
```

4. After the framework startup loading finishes, send a notification to the framework that startup page is about to close.



5. Send a notification after the home page ViewController starts, and the startup page is actually closed.

ViewController is usually the rootViewController of the Launcher micro application. If the ViewController is a TabBarController , it is the ViewController where the first tab is located.

```
@implementation HomeViewController
- (void)viewDidLoad {
    [super viewDidLoad];
}
- (void)viewDidAppear:(BOOL)animated {
    [super viewDidAppear:animated];
    // Notify the Launcher that home page has been displayed.
    [[NSNotificationCenter defaultCenter]
postNotificationName:@"kNotificationLauncherDidAppear" object:nil];
    }
    @end
```

## **Related links**

For more information about the startup page APIs, see API description.

# 5.4. Configure preset booth

To preset a booth on an iOS page, you need to perform two operations: data access and UI access. Data access callback obtains booth data. UI access obtains UI objects generated based on booth data.

If you are a beginner, we recommend that you configure the advertisement booths of iOS pages dynamically on the server-side mPaaS console. For details, refer to Create a booth.

ONOTE The access object in booth preset is usually a subclass of ViewController, and must implement the PromotionCenterDelegate method.

#### **UI access procedure**

1. Add protocol implementation.

The ViewController object implements the delivery protocol PromotionCenterDelegate .

```
@interface DemoViewController () <CDPPromotionCenterDelegate>
@end
```

#### 2. Add delivery listener.

Call the added listener operation in or before the viewDidLoad of the ViewController . When the delivery data is ready, the API will be called back through the protocol method.

3. Implement the callback method of the spaceView object in the PromotionCenterDelegate protocol to add the generated view to the page.

4. Remove the listener.

Remove the listener upon page closing. You can remove the listener in the dealloc method of ViewController .

```
- (void)dealloc {
    [CDPPromotionCenter removeObserver:self];
}
```

#### **Data access procedures**

1. Add protocol implementation.

The ViewController object implements the delivery protocol PromotionCenterDelegate .

```
@interface DemoViewController () <CDPPromotionCenterDelegate>
@end
```

2. Add delivery listener.

Call the added listener operation in or before the viewDidLoad of the ViewController . When the delivery data is ready, the API will be called back through the protocol method.

3. Implement the callback method of the spaceInfo object in the PromotionCenterDelegate protocol and perform custom processing based on the returned data.

}

4. Remove the listener.

Remove the listener upon page closing. You can remove the listener in the dealloc method of ViewController .



```
- (void)dealloc {
    [CDPPromotionCenter removeObserver:self];
}
```

# **5.5. API reference**

This topic describes the Mobile Content Delivery Platform (MCDP) related APIs and the startup page related APIs.

#### **Delivery APIs**

### **Delivery API class**

The delivery API class is the unified entry point for delivery functions. You can call the operations in the class to implement the related MCDP functions.

```
@interface CDPPromotionInterface : NSObject
@end
```

## Start component

#### **API** definition

+ (void)start;

### **API** description

You can call this operation to start the MCDP component, including initializing all delivery data and pre-adding resource pictures.

#### Example

[CDPPromotionInterface start];

#### **View MCDP component version**

## **API** definition

+ (NSString \*)sdkVersion;

## **API description**

You can call this operation to view the version number of the MCDP component.

Applicable versions: The API is available in version 3.0.0 and above.

#### Example

```
NSString cdpVersion = [CDPPromotionInterface sdkVersion];
```

# Report advertisement sharing events. API definition

+ (NSString \*)reportShareAdAction;

## **API description**

You can call this operation to report advertisement sharing events so that the sharing data can be displayed on the activity effect page of the MCDP console. The sharing data includes the sharing times and the number of sharing devices. The sharing times refers to the number of times that an advertisement is tapped and shared on the client page during a certain period of time. The number of devices with sharing operations refers to the total number of deduplicated devices on which the Share button is tapped on the client page during a certain period of time. For more information about advertisement sharing data, see View activity effect.

Applicable versions: The API is available in version 3.0.0 and above.

#### Example

CDPPromotionInterface reportShareAdAction];

## **Delivery properties**

Configure delivery properties, including logon notification name, logout notification name, and whether to use positioning information.

## **Property definition**

@interface CDPPromotionInterface : NSObject //The following are subject to user's configuration. @property (nonatomic, strong) NSString\* loginNotificationName; // The logon notificatio n name. This parameter has no default value. @property (nonatomic, strong) NSString\* logoutNotificationName; // The logout notificat ion name. This parameter has no default value. @property (nonatomic, assign) BOOL useLbsInfo; // Whether to use location information, the default is NO.

@end

## **Property description**

Property	Туре	Required	Description
loginNotificationN ame	NSString	No	The logon notification name. This parameter has no default value. MCDP will refresh the content delivered to the corresponding user according to the logon notification.

Property	Туре	Required	Description
logoutNotification Name	NSString	No	The name of the logout notification. This parameter has no default value. MCDP will clean up the cached delivery data according to the logout notification.
useLbsInfo	BOOL	No	Whether to use location information inside MCDP, the default is NO, representing not to use location information.

#### Example

[CDPPromotionInterface sharedInstance].loginNotificationName = @"xxxx";

## Startup page APIs

MCDP provides a series of methods in C programming language to execute the logic related to the startup page, which includes:

- 1. Determine whether to display the startup page.
- 2. Open the startup page.
- 3. Send notification that the startup page will close.

#### Determine whether to display the startup page

#### **API** definition

BOOL splashScreenExist(BOOL shouldShowSplashScreen);

## **API description**

Determine if the startup page needs to be displayed based on user-defined rules and MCDP internal rules.

#### **Parameters**

Parameter	Туре	Required	Description
shouldShowSplas hScreen	BOOL	Yes	Determine if the startup page needs to be displayed based on user-defined rules.

#### Example

BOOL shouldShowSplash = YES;
// Whether to display advertisement on the startup page.
shouldShowSplash = splashScreenExist(shouldShowSplash);

## Open the startup page

#### **API definition**

```
UIWindow * APSplashScreenStart(void (^dismiss)(void));
```

## **API description**

Open the startup page and display the startup page advertisement.

#### Parameters

Parameter	Туре	Required	Description
dismiss	block	Yes	Callback block of closing the startup page.

#### **Returned values**

Parameter	Description
UIWindow	The created startup page window.

#### Example

```
UIWindow *splashWindow = APSplashScreenStart(^{
    // The callback of closing the startup page.
    // Do something.
});
```

# Send notification of startup page closing API definition

```
void APWillKillSplashScreen(void);
```

## **API description**

After the framework startup is loaded, notifies the framework that the startup page is about to close. After the notification, the startup page may not be closed immediately.

#### Example

```
APWillKillSplashScreen();
```

# **6.Integrate MCDP into HTML5** 6.1. Configure preset booth

The operation of preset booths on HTML5 page involves obtaining booth information only. The obtaining methods includes: Obtain one by one and Obtain in batches .

In addition to obtaining booth information, you can also intercept booth information and advertising information by calling the API operations.

### About this task

If you are a beginner, we recommend that you configure the advertisement booths of iOS pages dynamically on the mPaaS console (server side). For details, refer to Create a booth.

## Obtain one by one

Call getCdpSpaceInfo on the HTML5 page to obtain booth information:

```
AlipayJSBridge.call('getCdpSpaceInfo', {
  spaceCode: 'space-codel',
  extInfo: {
    tradeNo: '123'
  },
   immediately: false,
  multiCallback: true
}, function (result) {
   console.log(result);
});
```

#### where:

- spaceCode : The booth code, which is applied for to the backend.
- extInfo : The extension information, in the format of key/value.
- immediately : The boolean value. It indicates whether cache data is not required and data is directly obtained from the server. By default, data is read from the cache and does not need to be transferred.
- multiCallback : The boolean value. It indicates whether multiple callbacks are required when the API is called once.

#### ? Note

If multiCallback is set to NO or not set, callback is performed only once. If there is a cache mechanism, ads cannot be displayed as scheduled when the page is opened for the first time, and are displayed the next time when the page is opened. If this parameter is set to YES, you need to perform MCDP callback twice. The first callback is the cached data (if there is a cache), and the second callback is the result returned by the RPC.

#### **Obtain in batches**

Call getCdpSpaceInfos on the HTML5 page to obtain booth information.

```
AlipayJSBridge.call('getCdpSpaceInfos', {
   spaceCodes: ['space-code1', 'space-code2'],
```

9<sub>910集团</sub> Mobile Content Delivery Platfo <sup>ANTGROUP</sup> rm

```
extInfo: {
   tradeNo: '123'
 },
 immediately: false,
 multiCallback: true
}, function (result) {
 console.log(result);
});
/**
* MCDP advertising service API
*/
public abstract class CdpAdvertisementService extends ExternalService {
   /**
    * Initialize all advertisement information.
    * @param extInfo: extension information.
     * @param callBack: callback.
    */
   public abstract void initialized(Map<String, String> extInfo,
IAdGetSpaceInfoCallBack callBack);
   /**
    * Query the booth by booth ID. Call back onSuccess(SpaceInfo spaceInfo)
asynchronously to return the query result.
    * If no local cache exists, the RPC query result is called back.
    * If a local cache exists and it's before update time, the local query result is c
alled back.
    \, * If a local cache exists and it's before update time, the local query result is c
alled back, and the update result is then called back after the RPC.
    * The onFail() API is called back only if the RPC fails.
     * @param spaceCode: booth ID, which needs to be applied for to the advertisement d
elivery platform.
    * @param callback: callback API.
    */
   public abstract void getSpaceInfoByCode(String spaceCode,
IAdGetSingleSpaceInfoCallBack callback);
    /**
    * Query the booth by booth ID. Call back onSuccess(SpaceInfo spaceInfo)
asynchronously to return the query result.
    * If no local cache exists, the RPC query result is called back.
     \star If a local cache exists and it's before update time, the local query result is c
alled back.
    * If a local cache exists and it's before update time, the local query result is c
alled back, and the update result is then called back after the RPC.
    * The onFail() API is called back only if the RPC fails.
     * @param spaceCode: booth ID, which needs to be applied for to the advertisement d
elivery platform.
     * @param extInfo: extension parameter.
     * @param immediately: only return the RPC result.
    * Anorom collbook collbook NDT
```

```
" Uparam Calipack: Calipack API.
    */
   public abstract void getSpaceInfoByCode(String spaceCode, Map<String, String> extIn
fo, boolean immediately, final IAdGetSingleSpaceInfoCallBack callback);
    /**
    * Query booths in batches based on the booth ID list. Call
onSuccess(List<SpaceInfo> adSpaceInfo) asynchronously to return the query result.
    * If no local cache exists, the RPC query result is called back.
    * If a local cache exists and it's before update time, the local query result is c
alled back.
    * If a local cache exists and it's before update time, the local query result is c
alled back, and the update result is then called back after the RPC.
    * Callback onFail(ListString adSpaceCodes) API only when RPC fails
    * @param spaceCodeList: booth ID list, which needs to be applied for to the
advertisement delivery platform.
    * @param extInfo: extension parameter.
     * @param immediately: only return the RPC result.
     * @param callback: callback API.
    */
   public abstract void batchGetSpaceInfoByCode(List<String> spaceCodeList,
Map<String, String> extInfo, boolean immediately, final IAdGetSpaceInfoCallBack callbac
k);
    /**
    * TODO: reserved H5 API.
     *Query and display ads. It is called only by AdH5Plugin.
    * @param activity: current page.
     * @param parentView: parent control.
     * @param url: URL.
    * @param h5Param: parameter.
    */
   public abstract void checkAndShowAdInH5(final Activity activity, ViewGroup parentVi
ew, String url, String h5Param);
    /**
    * Remove the advertisement view of the specified spaceCode.
    * @param activity: page from which an announcement is to be removed.
    * @param spaceCode: ad ID
    */
    public abstract void removeAdvertisement(Activity activity, String spaceCode);
    /**
    * Obtain the action executor. If this parameter is not set, null is returned.
    * @return: action executor.
    */
   public abstract ActionExecutor getActionExecutor();
    /**
    \star Set the action executor.
```

```
* @param executor: action executor.
*/
public abstract void setActionExecutor(ActionExecutor executor);
/**
* Set the user ID.
*
* @param userId: user ID.
*/
public abstract void setUserId(String userId);
/**
* Obtain the user ID.
 *
* @return: user ID.
*/
public abstract String getUserId();
/**
* Obtain callback classes for advertisement booths.
*/
public interface IAdGetSpaceInfoCallBack {
   /**
    * Succeeded in obtaining the booth information.
    * @param adSpaceInfo: booth list.
    */
    void onSuccess(List<SpaceInfo> adSpaceInfo);
    /**
    * Failed to obtain booth information.
    * @param adSpaceCodes: code list of booth requests.
    */
   void onFail(List<String> adSpaceCodes);
}
/**
* Gets callback classes for a single booth.
*/
public interface IAdGetSingleSpaceInfoCallBack {
   /**
    * Succeeded in obtaining the booth information.
    * @param spaceInfo: booth information.
    */
    void onSuccess(SpaceInfo spaceInfo);
    /**
    * Failure
    */
    void onFail();
}
```



}

#### ActionExecutor

ActionExecutor (action executor) implements the interception of booth and advertisement information. The client does not display the corresponding booths and advertisement content after the interception, and displays the booths and advertisement content without the interception.

```
/**
* Action executor
 *
*/
public interface ActionExecutor {
    /**
    * Whether to intercept the action
    * @param spaceInfo: booth information.
    * @param spaceObjectInfo: ad information.
     * @param url: action url.
    * @return: The value true indicates intercepting the action whereas false indicate
s otherwise.
    */
   boolean interceptAction(final SpaceInfo spaceInfo, final SpaceObjectInfo
spaceObjectInfo, final String url);
    /**
    * Execute the action.
     *
    * @param spaceInfo: booth information.
    * @param spaceObjectInfo: ad information.
     * @param url: action url.
     * @return: Value 1 indicates success and other values indicate exceptions.
    */
   int executeAction(final SpaceInfo spaceInfo, final SpaceObjectInfo spaceObjectInfo,
final String url);
}
```

# **7.Console operations** 7.1. Manage group

# 7.1.1. About user group

The group management function provided by the MCDP component enables you to accurately find the target audience based on business scenarios, which improves the effectiveness of advertising delivery. Before you create an activity, you need to create a target group for the activity. When you create an activity, you can directly call the created group as the target audience of the activity. The created group can be reused in different marketing activities to facilitate rapid marketing.

You can create a group based on user ID or device ID.

- **User ID**: Delivers activities to user IDs (userId) in the group. When a user ID logs in the App, the App displays the delivered content (booth + material/message push) according to the activity configuration. When the same user ID logs in different devices, the App always displays the delivered content.
- **Device ID**: Delivers activities to the device IDs (utiid) in the group. When a user opens the App on an assigned device, the App displays the delivered content (booth + material/message push).

In the process of creating a group, you can add a group in the following two ways:

- **Input directly**: You can directly enter user IDs or device IDs on the operation page. You must separate each piece of data with an English comma (,) or a line break. Each piece of data must be within 100 characters. The data exceeding 100 characters is not allowed.
- Import file: You can add a group ID by importing a file. When you add user IDs or device IDs to the file, you must separate each piece of data with a line break. Each piece of data must be within 100 characters. The data exceeding 100 characters is not allowed. The file size must be less than or equal to 50 MB, and the file must be a .txt file. When you import the ID, the system automatically deduplicates the ID.

# 7.1.2. Create a group

Log in to the mPaaS console, select the target app, and perform the following steps to create a user group:

- 1. On the left-side navigation pane, choose **Mobile Content Delivery Platform > Group management**, and click **Create a group**.
- 2. In the **Create a group** page, enter the group name and choose the group type.
  - Group name: Must be unique. Up to 30 characters are supported.
  - **Group type**: Choose **User ID** or **Device ID** based on your needs. You cannot modify the group type after submission.
- 3. Add the group by ID. You can add user IDs or device IDs by inputting them directly or importing a file containing the IDs.
  - Input directly: Enter one or more IDs, separated by commas (,) or line breaks. Up to 100 characters are supported. Only the first 100 characters will be imported for entries exceeding the limit.



- Import file: Click the **Click to upload** button and select the file containing the IDs. The file must be a TXT file within 50 MB, and can contain a maximum of 100 characters. Only the first 100 characters will be imported for entries exceeding the limit. The IDs will be deduplicated automatically after being imported.
- 4. Click Save to submit.

After submission, the page jumps to **Group management** page automatically, and you can view the added group information on this page. For more information, refer to Manage groups.

# 7.1.3. Manage groups

The group list shows all the successfully created groups. You can search for the target group, view and edit the group information, export user/device IDs in the group , or delete the group directly.

#### View the group list

Log in to the mPaaS console, select the target application, click **Mobile Content Delivery Platform**> **Group management** in the left navigation bar, and navigate to the **Group management** page to view the group list.

The group information displayed in the list includes:

- Group name: The name of the group.
- **Group ID**: It refers to the ID automatically generated after the group is successfully created. The group ID uniquely identifies a group.
- Users/Devices: The number of user IDs or device IDs contained in the group.
- **Group type**: It can be divided into two types: **User ID** or **Device ID**. For the advertising effects of people in different types, please refer to Create a group.
- **Creator**: Create an account for the corresponding group.
- Creation time: The creation time of the group, accurate to the second.
- **Update time**: The update time of the group, accurate to the second. Updated refers to edit or delete the group ID. If there is no update for the group, the update time is displayed as the creation time.
- **Status**: Group status can be **Ready** or **Uploading**. The meaning of each status is as follows:
  - **Uploading**: Indicates that the user/device information of the group is being uploaded.
  - **Ready**: It means that the user/device ID input has been done. Only groups in this status can be called by marketing activities.
- **Operation**: Through the **Operation** column, the operations of Edit group, Export user/device ID, and Delete group can be realized.

#### Search group

You can search for groups based on group type, keywords of group name or ID. The procedure is as follows:

- Group type: You can choose All group types, User ID or Device ID .
- Keywords of group name/ID: Input the Group name or the keywords of Group ID, and click the Search button.

The above search conditions can be used in combination. The system will search the group list according to the selected type and keywords, and display the group information that meets all the selected conditions in the list.

#### View group details

You can view the user IDs or device IDs contained in the group, and delete the target ID. The procedure is as follows:

- 1. In the group list, click the group name of the target group to enter the **Group details** page of that group.
- The User/device ID information list displays all the ID information contained in the current group. Click Delete in the Operation column corresponding to the target ID to delete the group ID.

#### **Edit group**

You can edit group information, including modifying group information and adding group ID. The procedure is as follows:

- 1. In the group list, click **Edit** in the **Operation** column of the target group to enter the **Modify group** page.
- 2. Modify the group name, and add users or devices.

Group type cannot be modified, and the user/device IDs contained in the group are not displayed on the **Modify group** page. You can check them on the **Group details** page, see View group details.

For how to add users or devices in the group, see Create a group.

#### Export user/device ID

You can export the group ID contained in the group by one click.

In the group list, click **Export** in the **Operation** column corresponding to the target group to export a .txt file containing user/device IDs.

#### **Delete group**

You can delete a group by a click, the procedure is as follows:

#### ? Note

User groups applied in activities cannot be deleted.

- 1. In the group list, click **Delete** in the **Operation** column corresponding to the target group.
- 2. In the pop-up prompt box, click the **OK** button to confirm the deletion.

# 7.2. Booth management

# 7.2.1. About booth

A booth is a rectangular area on a client page used to display advertising materials.

#### **Booth type**

MCDP supports the following seven types of booths: splash, pop-up, bulletin, banner, list, float, and custom booth.

- **Splash**: A booth in a full-screen area when the app is cold boot. And the splash booth is globally unique within the app.
- **Pop-up**: A booth in the app with controllable width and height in a floating layer area.
- Bulletin: Deliver relevant information to users in the form of message broadcasting.



- **Banner**: A booth in a rectangular area of a fixed position on the app page that supports a horizontal banner of one or more advertisement spaces.
- List: Many different ads are listed in an area on the app page.
- Float: A booth at a fixed position on the app page.
- Custom: Custom advertisements of copy type.

#### **Basic booth information**

The booth has basic information such as booth name, booth ID, and effective (cache) time.

- **Booth name**: Required. You can enter a maximum of 30 characters, and the name cannot be repeated with an existing name.
- **Booth ID**: Required. The ID can contain letters, numbers, underscores (\_), and hyphens (-), and cannot be the same as the existing booth ID.
  - **Custom ID**: If the booth is a dynamic booth, select this field, then enter the booth ID and page address in the **Custom ID** text box.
    - iOS page address: Enter the iOS-client page address which displays the booth. To enter the VCName of the page, you need to contact the app developer, for example, MPHomeViewController
       Note that VCName must inherit from DTViewController
       Otherwise, the booth will not be displayed.
    - Android page address: Enter the Android-client page address which displays the booth. To enter the ActivityName of the page, you need to contact the app developer, for example, com.mpaas.demo.cdp.DynamicContent1Activity
    - HTML5 page address: Enter the HTML5 page address which displays the booth. To enter the full path of the page, you need to contact the app developer. You can use the window.location
       object to get the full path of the HTML5 page.
  - Client preset ID: When the booth is a client preset booth, select this field, and copy and enter the ID of the preset booth in the Client Preset ID text box. You can contact the app developer to obtain the ID.
  - Splash booth ID: Select this field when it is a splash booth. The default splash booth ID is STARTPAGE and cannot be modified. An application can have only one splash booth, that is, if there is already one splash booth in the application, you cannot create another splash booth.
- **Valid (cache) period**: Required. Refers to the retention time of the cache after the booth advertisements are loaded once. The time should be set in consideration of the usage of download traffic and user experience. The time range that can be set ranges from 00:00:00 to 71:59:59.

#### **Booth configuration information**

Booths can be configured for booth type, booth material type, booth position, style, booth height, booth width, and the number of advertisement spaces.

- **Booth type**: The MCDP supports seven booth types: splash, popup, bulletin, banner, list, float, and custom booth.
- **Booth material type**: Required. Refers to the types of materials supported by the booth. Some booths can be adjusted.
- **Booth position**: Required. Refers to the position of the booth on the page, including the top floating layer of the page, the top of the page, the bottom of the page, the head of the list, the bottom of the list, the top left of the screen, the top right of the screen, the middle left of the screen, the middle right of the screen, the bottom left of the screen, and the bottom right of the screen. Some booths can be adjusted.

#### ? Note

Different booth types can be configured with different positions. For details, see the following booth configuration information sheet.

Top float	Top of page	Bottom of page	List head
The booth is located at the top of the page, floating above the page layout, and does not move down the original layout. It is a fixed position relative to the screen, the width of the booth is the same as the width of the page.	The booth is located at the top of the page and moves down the original layout. It is a fixed position relative to the page, the width of the booth is the same as the width of the page.	The booth is located at the bottom of the page and moves up the original page layout. It is a fixed position relative to the page, the width of the booth is the same as the width of the page.	Located at the top of the list (above the first advertisement space), and moves down the original list layout to the bottom. It is a fixed position relative to the list, the width of booth is the same as the list width.
List bottom	Upper left of screen	Middle left of screen	Bottom left of screen

Located at the bottom of the list (below the last advertisement space), and moves up the original list layout to the top. It is a fixed position relative to the list, the width of the booth is the same as the width of the list.	Only for float booth. Located at the top left of the screen, floating above the page layout, does not affect the original page layout, and is a fixed position relative to the screen.	Only for float booth. It is located in the center of the left side of the screen, floating above the page layout, does not affect the original page layout, and is a fixed position relative to the screen.	Only for float booth. Located at the bottom left of the screen, floating above the page layout, does not affect the original page layout, and is a fixed position relative to the screen.
Upper right of screen	Middle right of screen	Bottom right of screen	



- **Style**: Required. That is, the appearance and style of the booth, including background color, foreground color, text color, and control color. Some booths can be adjusted.
  - **Background color**: The background color of the booth, the value is **#F8F8F8** by default.
  - Foreground color: The foreground color of the booth. The foreground color of the booth is the background color that appears when the booth is clicked and the value is #E7E7E5 by default. When the booth is clicked, if there is a redirection address for the materials in the booth, the foreground color is rendered. Otherwise, the background color is rendered. For the configuration of the redirection address, see Creating an activity > Adding advertising.
  - **Text color**: The color of the text in the booth, the value is **#FF0000** by default.
  - **Control color**: The color of the control (small horn) in the booth, the value is #1A1311 by default.
- **Booth height**: Required. The height of the booth on the application page in px. Some booths can be adjusted.
- **Booth width**: Required. The width of the booth on the application page in px. Some booths can be adjusted.

• **Advertising spaces**: Required. Booths of banner and list type support multiple frames, and the number of advertisement spaces is the number of frames in the booth. Some booths can be adjusted.

Ad space 1	Ad space 1	Ad space 2
Ad space 2 Ad space 3	Ad space 3	Ad space 4

• **Display duration of each message**: This configuration is associated with the number of advertisement spaces. The carousel time of single information is not configurable when the number of advertisement spaces is 1. And when the number of advertisement spaces is not 1, the carousel time of single information can be configured as an integer from 1 to 10.

	Spla sh	Pop -up	Bulletin	Banner	List	Float	Cus tom
Booth Type					$\checkmark$		$\checkmark$
Booth material type	V	V	x	$\checkmark$	$\checkmark$	$\checkmark$	x
			Top float	Top float	Top float	Upper left of screen	

			Top of page	Top of page	Top of page	Upper right of screen	
			Bottom of page	Bottom of page	Bottom of page	Middle left of screen	-
Booth position	Def ault	Def ault posi tion	List head (Only available for Native page)	List head (Only available for Native page)	List head (Only available for Native page)	Middle right of screen	x
			List bottom (Only available for Native page)	List bottom (Only available for Native page)	List bottom (Only available for Native page)	Bottom left of screen	
			-	-	-	Bottom right of screen	
Style	x	x	$\checkmark$	x	x	x	$\checkmark$
Booth height	x	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	x
Booth width	x	$\checkmark$	x	x	x	$\checkmark$	x
Advertising spaces	1	1	1	1-10	1-10	1	1
Display duration of each message	x	x	x	1-10s	1-10s	x	x

**Note**: " $\sqrt{}$ " means this element is supported and must be configured; "x" means this element is not supported; "-" means no content.

### Supported booth materials

The types of booth materials supported by MCDP are listed in the following table:

- **Text**: Plain text is displayed.
- **Static image**: Static images in PNG, JPG, and JPEG formats are displayed.
- **GIF**: GIFs are displayed.

- Lottie animation: Lottie animations are displayed.
- **HTML5 page**: Online HTML5 pages are displayed.

The types of booth materials supported vary by booth type, as detailed in the following table.

	Splash	Pop-up	Bulletin	Banner	List	Float	
Static image			x	$\checkmark$		$\checkmark$	
GIF	$\checkmark$	$\checkmark$	x	$\checkmark$		$\checkmark$	
Lottie animatio n	$\checkmark$	$\checkmark$	x	$\checkmark$		$\checkmark$	
HTML5 page	x		x			x	
Text	x	x	$\checkmark$	x	x	x	
⑦ Note "√" means supported; "x" means not supported.							

#### **Booth configuration modes**

MCDP provides two booth configuration modes: **dynamic configuration** and **client preset**. You can choose one as needed.

#### • Dynamic configuration

In the case of dynamic booth configuration, after you added the MCDP SDK to the client project, directly configure booth properties on the console, including the booth name, booth ID, booth page, booth type, booth material type, booth position, booth width, booth height, and the amount of advertisement spaces.

The booth types supported by a dynamic booth include pop-ups, bulletins, banners, lists, float, and custom booth. For the booth positions supported by various types of booths, see the Booth position column in booth configuration information sheet. In dynamic configuration, only the content of the booth is configured, not the position. Dynamic booths do not support the belly-band booth position.

Take note of the following items when you use the dynamic booth:

 For a custom booth, you need to preset the booth on the client first. This type of booth is special and applies to partially scenarios where custom copywriting is required. Take the search box as an example, on the console, you can define the content of the copy in the search box, but you cannot set the position, width, and height of the search box, and these properties need to be configured through the client.

- In Android client, the top floating layer or the top of the page must contain the ViewGroup control with tag h5\_titlebar, or the layout must contain the AUTitleBar control from the antui-build component; the head/bottom of the list must contain the ListView control.
- In iOS client, the head/bottom of the list must contain the UITableView control.

#### • Client preset

In the case of booths preset on the client, you need to add the MCDP SDK to the client project and preset some of the booth properties in the client project by coding, including the ID, page, position, width, and height of the booth. Then you need to create the booth on the console and enter the booth ID preset in the client project to complete the booth creation. You can also set the booth width and booth height on the console, but the two can only be set when they are not configured in the client project.

The types of booths supported by booths preset on the client include pop-ups, bulletins, banners, lists, float, and custom booth. The client configuration mode is available to support booth positions that are not supported by dynamic booths (for example, belly-band position).

? Note

For ease of adjustment, we recommend that you preset only the ID, page, and position of the booth in the client project and configure only the booth height on the console.

For details about the procedures of presetting a booth on different pages, refer to the following articles:

- Android page: Configure preset booth in Android app.
- **iOS page**: Configure preset booth in Android app.
- **HTML5 page**: Configure preset booth in HTML5 page.

# 7.2.2. Create a booth

With the booth management function provided by MCDP, you can customize and manage the booths to effectively control the traffic. Before delivering an advertisement, create a booth in advance, specify the client page, position, type, material type, and height of the booth as needed, so that when you create a marketing campaign, you can directly select the booth and add the corresponding advertisement content or elements. Each created booth can be reused in different advertisement deliveries to facilitate marketing campaigns.

According to the different configuration modes, booths can be divided into the following three categories:

#### • Dynamic booth

After you added the MCDP SDK to the client, directly configure all the booth properties on the console, including the ID, page, display style, position, material, and height of the booth, without any operation on the client.

Dynamic booths do not support configuring a booth in the center position.

#### Client preset booth

After you added the MCDP SDK to the client, preset some of the booth properties on the client by coding (including the ID, page, position, and height of the booth), then create a booth on the console, and enter the booth parameters preset on the client. When the information configured on the console does not match the information configured on the client, the actual booth information on the client shall prevail.



- Applicable scenario: The client configuration mode is available to support booth positions that are not supported by the dynamic configuration mode (for example, the center position).
- Recommended mode: For ease of adjustment, the ID, page, and position of the booth are preset on the client, and all other booth properties are configured on the console.
- Splash booth

A special preset booth. Only one splash booth can be created in an app, and by default, the splash booth ID is STARTPAGE and cannot be modified.

#### Create a dynamic booth

#### **Prerequisites**

- MCDP SDK has been added to the client.
- For a custom booth, you need to preset the booth on the client first. This type of booth is special and applies to partially scenarios where custom copywriting is required. Take the search box as an example, on the console, you can define the content of the copy in the search box, but you cannot set the position, width, or height of the search box, these properties need to be configured through the client.
- In Android client, the top floating layer or the top of the page must contain the ViewGroup control with tag h5\_titlebar, or the layout must contain the AUTitleBar control from the antui-build component; the head/bottom of the list must contain the ListView control.
- In iOS client, the head/bottom of the list must contain the UITableView control.

#### **Procedures**

Log in to the mPaaS console, select the app, and perform the following steps to create a booth:

1. On the left navigation bar, enter **Mobile Content Delivery Platform** > **Resource management** >

Booth management page, click Create a booth, and enter Basic information page.

- 2. Enter the booth name, select **Custom ID**, enter the page address, and then set the valid (cache) time.
  - iOS page address: Enter the iOS-client page address of the booth to be displayed. To enter the VCName of the page, you need to contact the app developer, for example, MPHomeViewController
  - **Android page address**: Enter the Android-client page address of the booth to be displayed. To enter the ActivityName of the page, you need to contact the app developer, for example, com.mpaas.demo.cdp.DynamicContentlActivity.
  - **HTML5 page address**: Enter the HTML5 page address of the booth to be displayed. To enter the full path of the page, you need to contact the app developer. You can use the window.location object to get the full path of the HTML5 page.
- 3. Click **Next** to enter **Configuration details** page, and configure the booth details. For the description of each parameter, see Booth configuration information.
- 4. After confirming all the information, click **Submit**.

After completing the booth creation, you can view the information of the booth you have just created in the booth list, edit and delete the booth. For operation details, see Booth management > Booth list.

#### Create a client preset booth

> Document Version: 20231226

#### Prerequisites

In the case of booths preset on the client, you need to preset the booths by coding in the client project, and record the ID, page, and position of the preset booth, and only configure the booth height on the console. For booth preset operations, see Preset booth (iOS) and Preset booth (Android).

#### Procedures

Log in to the mPaaS console, select the app, and perform the following steps to create a booth:

1. On the left navigation bar, enter **Mobile Content Delivery Platform** > **Resource management** >

**Booth management** page, click **Create a booth**, and enter **Basic information** page.

- 2. Enter the booth name, select **Client preset ID**, enter the client preset booth ID, and set the valid (cache) time.
- 3. Click **Next** to enter **Configuration details** page, and configure the booth details. For the description of each parameter, see Booth configuration information.
- 4. After confirming all the information, click **Submit**.

After completing the booth creation, you can view the information of the booth you have just created in the booth list, edit and delete the booth. For operation details, see Booth management > Booth list.

#### Create a splash booth

#### **Prerequisites**

MCDP SDK has been added to the client and the startup page (Configure an iOS startup page/Configure an Android startup page) has been configured.

#### **Procedures**

Only one splash booth can be created in an app.

Log in to the mPaaS console, select the app, and perform the following steps to create a booth:

1. On the left navigation bar, enter **Mobile Content Delivery Platform** > **Resource management** >

Booth management page, click Create a booth, and enter Basic information page.

2. Enter the booth name, select **Splash booth ID**, and then set the valid (cache) time.

? Note

If a splash booth has been created in the current app, the **Splash booth ID** tab will be unavailable.

- 3. Click **Next** to enter **Configuration details** page, and select the type of material and the amount of advertisement space. For the description of each parameter, see Booth configuration information.
- 4. After confirming all the information, click **Submit**.

#### Important

The splash booth provides only one advertising space.

After completing the booth creation, you can view the information of the booth you have just created in the booth list, edit and delete the booth. For operation details, see Booth management > Booth list.

# 7.2.3. Manage booths

You can view all the booths, booth details, and the application of the booths in the current App, modify and delete the booths.

#### **Booth list**

Information about all created booths is displayed in the **Booth list**. You can search for the target booth, view and modify the booth information.

Booths in the booth list are sorted in descending order by **Update time**. The information displayed in the list includes:

- Booth name
- Booth ID
- Type
- Material type
- Creation time
- **Update time**: Refers to the modification of booth information or the change of booth status.
- **Status**: The application status of the booth, including **Not in use**, **In use**, and **To be used** (The "to be used" status may occur when the activity has not yet been activated or has been deactivated). The three statuses are defined as follows:
  - **Not in use**: Indicates that the booth is not called by any activity at present. In this status, the booth can be edited and deleted.
  - **In use**: Indicates that the booth is being called by an activity and the activity has been activated or not yet been activated. For the description of the activity status, see View activity list. In this status, the booth cannot be edited or deleted.
  - **To be used**: Indicates that the booth was called by an activity, but the activity has gone offline. In this status, the booth can be edited but not deleted.

#### ? Note

When the activity launches and has been activated, the associated booth resources are called to display the advertising materials of the activity. For how to associate booths in an activity, see Create an activity > Add a booth for delivery.

• **Operation**: With the **Operation** column, you can view booth details, edit booths, and delete booths.

#### Search for booths

You can search for the target booth in the booth list according to the type, material type, name, or ID keyword of the booth. The procedures are as follows:

- 1. In the search box at the upper-right of the list, select booth type, booth material type, or enter the booth name/ID keyword.
- 2. Click **Search** to search for all the eligible booths in the list.

#### View booth details

> Document Version: 20231226

The booth list provides part of the information about the booths. For more information, please go to the **Booth details** page to view. The procedure is as follows:

In the booth list, click **Details** in the **Operation** column of the target booth and enter the **Booth details** page to view the details of the booth.

The information displayed on the **Booth details** page consists of two categories: **Basic information** and **Booth information**.

- **Basic information** includes:
  - Booth name
  - Booth ID: Includes booth ID and ID type, where **Custom** means the ID is a Custom ID; Client preset means the ID is a Client Preset ID.
  - Booth page address: This information is only available for dynamic booths. To obtain the page address information for preset booths, please contact the App developer.
- Booth information includes:
  - Booth type
  - Booth material type
  - Booth position: When it is a preset booth, if the booth position has been specified in the client configuration, the information configured on the client shall prevail and you need to contact the App developer to get the actual position of the booth.
  - Amount of advertisement spaces
  - Booth preview
  - Others: Different types of booths have different booth properties and the different booth information is displayed on the page. For example, bulletin booths have the **Style** property while banner booths have the **Carousel time of a single message** property.

#### Edit a booth

You can edit the booth information. The procedures are as follows:

NoteBooths in In use status cannot be edited.

- 1. In the booth list, click **Edit** in the **Operation** column of the target booth to enter the **Modify booth** page.
- 2. To modify booth information, please refer to About the booth for parameter descriptions.

Booth ID type and booth ID cannot be modified.

If the booth type is modified, the advertising spaces will automatically be changed to 1 and you need to reconfigure the amount of advertisement spaces.

#### **Delete a booth**

You can delete the booth in the booth list. The procedures are as follows:

? Note

Booths in In use and To be used status cannot be deleted.

- 1. In the booth list, click **Delete** in the **Operation** column of the target booth.
- 2. In the pop-up window, click **OK**.

#### **Booth application overview**

Click **Booth App overview** to enter the **Booth App overview** page. This page displays the application of the booth, i.e., information about the activities calling the booth.

The **Booth App overview** page is divided into the following three areas:

- Search area: Provides search criteria for you to search for booths.
- Booth list: Displays the names of all created booths. Click the booth name to select the booth. The "In use" label in the upper left corner of the booth name indicates that the booth is associated with a activity and the associated activity has not gone offline.
- App overview: Displays basic information about the selected booth, as well as the activities associated with the booth on the currently selected date.

#### Search for booths

**Booth overview** In the search area at the top of the page, three search conditions are provided as follows for you to search for the target booth:

- **Time**: Select the time interval to filter the booths called by the activity during the selected time interval. The current date is selected by default, and the time is from 00:00:00 to 23:59:59.
- Activity unit: Select the activity unit to filter the booths called by the selected activity unit. You can select one of the created activity units/all activity units.
- **Booth**: Enter the booth name or booth ID keyword to filter the booths whose names or IDs contain the keyword.

The preceding search criteria can be used in combination. In the **Booth list**, the background color of the booths that match the search criteria is blue, as shown in the following image.

#### View booth application overview

The procedures are as follows:

1. Click the booth name in the **Booth list** and select the target booth.

Only one booth can be selected at a time, and the selected booth will have a check mark displayed on the right of its name.

2. View the query results in the **Activity list**.

The **Activity list** shows information about the activities that called the currently selected booth during the selected date/month.

# 7.3. Material material

# 7.3.1. Create material

MCDP provides a material library for you to centrally manage various materials. When you create a marketing activity, you can directly use the resources in the material library as advertising materials to simplify the procedure.

This topic describes how to complete the material creation and explains relevant parameters during the creation.

Log in the mPaaS console, select the target application, and perform the following steps to create materials:

- 1. In the left navigation bar, select **Mobile Content Delivery Platform > Resource management > Material management** to enter the Material management page.
- 2. Click the **Create material** button to configure the material information.
- **Material name**: You can enter a maximum of 30 characters, and the entered name must not be duplicated with the name of an existing material.
- **Material type**: The supported material types include static images, GIFs, and Lottie animations.
- Booth type: Select the booth type according to the material. For example, you can select banner, which means the material will be applied to the banner booth. All the above three types of materials can be applied to splash, pop-up, banner, list, and floating icon booths. When you create an activity, you can use the three types of materials as the advertising materials for a banner booth. For more information about the instruction on how to add advertisement materials, see Create an activity > Add a delivery booth.
- 3. Upload a material. Click **Upload material** to open the file browser, choose the material you need to upload according to the selected material type, and then click **Open**.
  - Descriptions about the material size:

The maximum size for all the types of materials is 300 KB. Otherwise, you cannot upload the material.

- For static images, the file extension can be .png , .jpg , and .jpeg . The recommended size is 1080 px × 1920 px.
- For GIF images, the file extension can be .png , .jpg , and .jpeg . The recommended size is 1080 px × 1920 px. For Lottie animations, the material file can be a .json file generated by a Lottie animation. The file contains the coordinates of the key points and the motion trajectory of the animation. Depending on the booth type to which the material will be applied, we recommend uploading materials that vary in length, width, and size.
- Delete a material:

If you need to delete the selected material, you can hover the mouse on the file name under the preview window of the material. And then click the icon "X" on the right to delete the material.

4. Click **Submit** to save the material.

When a **Complete** notification appears on the page, the notification indicates that you complete the creation of materials. In the **Material list**, you can view information about the material you just created.

# 7.3.2. Manage materials

All added materials are displayed on the **Material management** page. You can search for the target material, modify or download the material.

Log in to the mPaaS console, select the target app, and enter the enter the **Mobile Content Delivery Platform > Material management** page from the navigation pane .

The materials are displayed in descending order by the creation time. Each piece of material shows the following information:

- Material name
- **Material ID**: A unique ID used to identify the material. Generated automatically by the system after the material is added.
- **Thumbnail** : Material thumbnail image. You can move your mouse cursor over the image to preview the material.
- Material type
- Booth type
- Creator: the account of the user who added the material.



- **Creation time**: The time when the material was created.
- Update time: The latest time that the material was edited.
- **Operation**: Offers the options of editing or downloading material.

# Search for a material

You can filter the materials by booth type and material type, and search for a material by material name or material ID. The procedure is as follows:

- 1. In the **Material management** page, enter the keyword in the search box in the upper right of the page. You can also filter the materials by booth type and material type first.
- 2. Click the Search button (  $\bigcirc$  ). The materials whose Material name or Material ID

contains the keyword are displayed on the page.

## Edit a material

You can modify the material information. The procedure is as follows:

- 1. In the Material management page, click the Edit button of the target material.
- 2. Modify the material information. For details about the procedure and parameters, see Create a material.

# **Download a material**

In the **Material management** page, click the **Download** button of the target material.

# 7.4. Manage activaty

# 7.4.1. About activity unit

Activity unit refers to a collection of marketing activities created around a certain marketing purpose. There can be multiple activities with the same theme under a activity unit. In MCDP, marketing activities are managed through activity units.

Each activity unit contains the following information:

- Activity unit name: The name of the activity unit. Enter it when creating a activity unit.
- Activity unit ID: The number of the activity unit. It is automatically assigned by the system when the activity unit is created.
- **Validity period**: The validity period of a activity unit depends on the start time and end time of the activities contained in the activity unit. When a activity unit contains multiple activities, the validity period of the activity unit starts at the earliest start time of the activities and ends at the latest end time of the activities.
- Creator: The creator of the activity unit.
- Creation time: The time when the activity unit is created in the system.
- **Activity status**: The independent status of each activity contained in the activity unit. There are three activity statuses: online, to be online, and offline.
- **Operations**: The operations that can be performed in the activity unit.
  - When all the activities in the activity unit are to be activated, or there are no activities in the activity unit, the activity unit can be edited or deleted.
  - When there are activities in the activity unit that are not to be activated (online or offline), you can only view the effects of the activity unit.

# 7.4.2. Create an activity unit

This topic introduces the procedures to create an activity unit.

# Procedures

Log on to the mPaaS console, select the target application, and perform the following steps to create an activity unit:

- 1. In the left navigation bar, select **Mobile Content Delivery Platform > Activity management**.
- 2. Click Create an activity unit, a dialog box will appear.

In the dialog box, configure the activity unit information:

- Activity unit name (required): A maximum of 30 characters is allowed and the name cannot be the same with an existing unit.
- Conversion indicator (optional): You can select event registered in Mobile Analysis Service (MAS). The occurrence of the event means that the activity conversion target is achieved. If no MAS event is selected, the conversion rate = the number of devices on which the activity is clicked / the number of devices on which the activity is browsed. For example, if you set the conversion indicator of a promotion marketing activity unit to logon event, when logon event occurs on the client, it means that the target of the activity unit has been achieved.
- 3. Click **OK** to save the information.

After the activity unit is created, the **Select the activity type** dialog box pops up. The activity creation starts and you can create an activity. For more information about the detailed steps to create an activity, see <u>Create an activity</u>.

# 7.4.3. Create an activity

This topic guides you to create a marketing activity. The whole process is divided into the following four steps:

- 1. Configure basic information
- 2. Configure user groups
- 3. Configure marketing events
- 4. Configure marketing actions

# About marketing activity

MCDP supports two types of marketing activities: **interactive marketing activity** and **active marketing activity**. Active marketing supports direct marketing and promotion activities for the targeted groups, i.e. advertisement delivery. Interactive marketing supports targeted marketing and promotion activities triggered only when the user performs a certain behavior or multiple behaviors, and two marketing actions, that is, advertisement delivery and message push, are supported. You can a marketing activity type as needed.

In an activity, MCDP supports two marketing delivery modes, advertisement delivery and message push. You can use materials such as text, images, or GIFs to deliver advertisements; or create a push message, and deliver the advertisements to the target consumers and push the message as soon as the activity takes effect.

# Prerequisites

Before creating an activity, make sure you have created the materials, booths, targeted groups, events, and activity units.

# Step 1: Configure basic information



- 1. Log on to the mPaaS console and select the app. In the left-side navigation pane, choose **Mobile Content Delivery Platform > Activity management**.
- 2. In the list of activity units, click the name of the target activity unit.
- 3. Click Add an activity and select the type of activity you want to create, i.e. Active marketing activity or Interactive marketing activity.
- 4. Configure the basic information of the activity.

The following table describes the **Basic information** parameters.

Parameter	Description
Activity Name	Required field. The name of an activity. You can enter at most 30 characters. The name of an activity in the same unit must be unique.
Activity type	<ul> <li>The type of the current activity.</li> <li>Active marketing supports direct marketing and promotion activities for targeted groups.</li> <li>Interactive marketing supports marketing and promotion activities for target users triggered when they perform one or more specific behaviors.</li> </ul>
Validity period	<ul> <li>Required field. The duration of an activity. You need to select the start time and end time, which can be accurate to seconds.</li> <li>Choose the date and time of the start and end time. The default value is the current time.</li> <li>In the time select box, select the start date and end date first, then click <b>Select time</b> and select the exact time by scrolling the bar.</li> <li>You cannot select a time earlier than the current date.</li> </ul>
Activity cycle	<ul> <li>The rule for periodically launching an activity. For example, launch an activity on a specific date. The options are described as follows:</li> <li>Daily: Launch an activity every day within the validity period of the activity.</li> <li>Weekly: Select the days in a week for launching an activity. You can select multiple consecutive or non-consecutive days in a week. For example, if you select Monday, Wednesday, and Thursday, the activity will only be launched on every Monday, Wednesday, and Thursday, and Thursday within the validity period of the activity.</li> <li>Monthly: Select the date for launching an activity. You can select multiple consecutive or non-consecutive days. For example, if you select 1st, 16th, and 17th, the activity will be launched only on the 1st, 16th, and 17th of each month within the validity period of the activity.</li> </ul>

	You can select event registered in Mobile Analysis Service (MAS). The occurrence of the event means that the activity conversion target is achieved. If no MAS event is selected, the conversion rate = the number of devices on which the activity is clicked / the number of devices on which the activity is browsed.
Conversion indicator	<ul> <li>The conversion indicator of an activity and of the unit to which the activity belongs can be different. The conversion indicator of each activity within a unit differs depending on the activity goal. For example, the goal of activity 1 is the occurrence of logon events, the goal of activity 2 is the occurrence of click events, and the goal of the activity unit is the occurrence of payment events. When viewing activity results, you can view conversion rate statistics of each activity and the activity unit based on the goals.</li> </ul>
	<ul> <li>Before associating an activity with MAS events, you must first create MAS events on the mPaaS Console &gt; Mobile Analysis Service &gt; Custom analysis &gt; Custom configuration page (see Configure events for instructions). Then you need to connect the MAS component to the client (see Access Android or Access iOS for instructions) and customize tracking points (see Tracking points on Android Clientor Tracking points on iOS Client for instructions).</li> </ul>
Activity description	The description of an activity. You can enter at most 100 characters.

5. After you configure basic information, click **Next**.

# Step 2: Configure user groups

On the **User groups** page, configure the target user group of an activity. MCDP supports two types of groups: **Target group** and **Whitelist group**. The following table describes the user group parameters.

Parameter	Description
Group	<ul> <li>The user groups created in the Group management menu is directly called. You can select either Target group or Whitelist group.</li> <li>When no user group is added, All groups (including the MAS group and MCDP group) are added by default.</li> <li>If you select both the target group and whitelist group, the union of the two types of the group will be applied.</li> </ul>
Platform	Required field. The platform for launching an activity. You can select <b>Android</b> , <b>iOS</b> , or both.
City	The city where an activity is launched. If this parameter is not configured, all cities are selected by default.

Version	<ul> <li>The version of the client for launching an activity. The version number consists of three digits, for example, 1.0.0.</li> <li>If this parameter is not configured, all client versions are selected by default.</li> <li>If you want to launch an activity on a client of a specific version, enter the version number in the minimum and maximum version number input boxes.</li> </ul>
	<ul> <li>This field applies for active marketing only. The extended parameters supports configuring targeting rules based on client parameters. For example, to push ads to users who use AA phones, enter mobileBrand in the parameter column, select the <b>equal to</b> condition, and enter <b>AA</b> in the value field. Currently, client built-in parameters include:</li> <li>osVersion: the system version, which is obtained from</li> </ul>
	Build.VERSION.RELEASE .
	• mobileBrand: the manufacturer, which is obtained from Build.BRAND .
Extended parameter	• mobileModel: the model, which is obtained from Build.MODEL .
	For Android clients, you can customize business extended parameters through one of the following two ways:
	• Use the second parameter in updateSpaceCode(String spaceCode, Map <string, string=""> extInfo, boolean immediately) of the preset booth.</string,>
	<ul> <li>Use the common extension method. For details, seeConfigure extended parameters of RPC requests.</li> </ul>
	For iOS clients, you can customize extension parameters by configuring the extInfo API.

The following sections describe how to add a target group and a whitelist group.

# Add a target group

- 1. Select **Target group** as the type of the group to be added.
- 2. In the group select box that appears on the page, click **+Add**.
- 3. On the **Select target groups** panel on the right, select the groups that have already been created. You can also click the **Create MCDP group** button in the upper-right corner add new group. For more information, see Create a group.

#### ? Note

The list in the panel displays the MAS groups and manually created groups. Through the search box above the list, you can search for the target group by the group source, and group name or group ID.

- 4. Select the target group from the list. You can select multiple target groups. After you finish selecting the target group, click **OK**.
  - If you add multiple groups to a group select box, the union of these groups is used as the target group of an activity.

- When you hover over a selected group in a group select box, the delete icon appears. You can click that icon to delete the selected group.
- 5. (Optional) Click the + **OR** button, and follow the preceding procedure to add one group or the union of multiple groups.
  - Each group select box is joined by the OR logic. That means, when multiple group select boxes are added, the union of all selected groups is used.
  - If there are two or more group select boxes, you can click the delete icon in the upperright corner to delete the corresponding select box.

# Add a whitelist group

- 1. Select Whitelist group as the type of the group to be added.
- 2. In the group select box that appears on the page, click + Add.
- 3. On the **Select whitelists** panel on the right, select the group that has already been created. You can also click the **Create MCDP group** button in the upper-right corner to add a new group. For more information, see Create a group.

NoteWhitelist group can only be a MCDP group.

- 4. Select the target whitelist from the list. You can select multiple whitelist groups. After you finish, click **OK**.
  - If you add multiple groups in a group select box, the union of these groups is used as the whitelisted group of an activity.
  - When target group and whitelist group are added at the same time, the union of all these groups is used as the group of an activity.

After you finish configuring groups, click **Next** to configure a marketing event.

# Step 3: Configure marketing events

No marketing events are required for active marketing activity.

On the **Marketing event** page, add a marketing event (preset MAS event) that triggers marketing actions.

Marketing events are required for creating interactive activities only. An interactive activity requires its marketing actions (booth placement or message push) to be triggered when a marketing event occurs for a user group of the activity. Therefore, you must configure a marketing event that triggers marketing actions.

Perform the following steps:

- 1. In the time duration select box, select a time period. You can only select a time period within the validity period of an activity.
- 2. Select a MAS event you created in advance from the select box after executed.
- 3. Select the condition for the MAS event to trigger marketing actions. You can select **Greater than**, **Equal to**, or **Less than**.
- 4. Select the threshold for the MAS event to trigger marketing actions. That is, the number of times a MAS event occurs.
- 5. Click + **Filter criteria** to add a filter condition for the MAS event to trigger marketing actions.

<sup>?</sup> Note



- i. Select the MAS event attribute for filtering from the Attribute drop-down list.
- ii. Select the condition to determine whether a filter condition is valid. You can select **equal to**, **not equal to**, **include** or **exclude**.
- iii. Select the value of the filter condition attribute.

? Note

You can add multiple filter conditions by using **OR** and **AND** logical operators. AND At most three filter conditions can be added under an **OR** or **AND** logic.

#### 6. Click Next.

## Step 4: Configure marketing actions

MCDP provides two types of marketing actions: **booth delivery** and **message push**. In active marketing activities, you can use the booth delivery marketing action. In interactive marketing activities, you can use the booth delivery, message push, or both types of marketing actions.

The following describes how to configure the two types of marketing actions. After you finish configuring marketing actions and submit them, a marketing activity is successfully created.

## **Booth delivery**

You can deliver the ads to multiple booths for a single activity. Complete the following operations to add a booth:

- 1. In the booth select area, click + Add a booth.
- 2. In the **Select a booth** panel on the right, select the booth you want to add and click **OK**.

#### ? Note

The list in the panel displays booths that are already created. Through the search box above the list, you can search for the target booth by the booth type, booth material type, and booth name/ID. You can instantly create a booth by clicking the **Create a booth** button. For details, see **Create a** booth.

3. The edit card of the selected booth is displayed on the page. Click **+Add material** to add materials (the advertising materials to deliver in the booth for an activity). Up to three sets of advertising materials can be added. The right side of the edit card provides a real-time preview of the booth, which refreshs in real time according to the added materials.

#### ? Note

When the activity material is an HTML5 material, the configured activity landing page cannot be redirected to.

You can add multiple materials for a booth. As the material type supported by different types of booths varies, the types of materials that are finally added to the activity also vary. For more information, see <u>About booths</u>.

4. Configure a fatigue control rule. Configure a fatigue control rule for materials displayed in the booth. Click Add a rule to add, multiple rules are supported. A booth contains one or more advertising spaces. And a single advertising space can display an activity material. For an activity that call booth resources, its materials can be displayed alternately in a booth according to fatigue rules.

The fatigue rule applies to all materials in a certain booth called by the current activity. That is, the two materials in the red box can be repeatedly displayed to the same user twice in a day. After the two materials have been all browsed two times in the booth by the same user, they will no longer be displayed when the booth is exposed to the user again on the same day.

Fatigue		Description	Supported booth types
	Always display	A material is always displayed within the validity period of a activity.	All booth types
Display	Disappear after certain times in a day	A material is no longer displayed after it was displayed for a certain number of times within a day. The number of display times can be customized.	Splash Bulletin
	Disappear after certain times	A material is not displayed after it was displayed for a certain number of times. The number of display times can be customized. Once: An advertising material is displayed in a booth when an app redirects to the page where the booth is located.	Splash Pop-up List Banner (multiple frames) Float Bulletin
	Disappear after certain times	The material is not displayed after it was clicked for a certain number of times. The number of clicks can be customized.	Splash Pop-up List Banner (multiple frames) Float Bulletin
Click	Disappear after certain times in a day	The material is not displayed after it was clicked for a certain number of times within a day. If the second day is still within the validity period of a activity, this material can still be displayed. The number of clicks can be customized.	Pop-up List Banner (multiple frames) Float Bulletin

#### The following table describes the fatigue rules supported by MCDP.

	Disappear after redirection	A material disappears and is not displayed after a user clicks it and is redirected to a specified Android, iOS, or HTML5 page of the booth.	Pop-up List Banner (multiple frames) Float Bulletin
	Disappear after certain times	A material is not displayed after it was closed for a certain number of times. The number of closing times can be customized.	Pop-up Bulletin Banner (single frame)
Close	Disappear after certain times in a day	A material is not displayed after it was closed for a certain number of times within a day. If the second day is still within the validity period of a activity, this material can still be displayed. The number of closing times can be customized.	Pop-up Bulletin Banner (single frame)
	Disappear immediately	A material disappears immediately and is not displayed after it was closed.	Pop-up Bulletin Banner (single frame)

#### Note:

- The **Display Always display** rule cannot be used with other rules. Otherwise, the rule will become invalid.
- If you add a Click Disappear after certain times or Click Disappear after certain times in a day rule, the number of clicks on a material is still be recorded even if no URL is configured for redirection of the material.

#### 5. Configure **Booth display priority**.

- Allocation rules of advertising space resources in the same booth. The following are the rules described according to priority:
  - Activity priority: Depends on the booth display priority, which is the priority of booth usage for the activity configured manually through the console. That is, activities with higher priority are displayed first.
  - Activity online time: When activities with the same priority compete for the same booth, the sorting logic is the activity freshness. That is, the latest online activities are displayed first.
  - Additional rules: The activities on the last day of their validity are displayed first. When multiple activities have the same priority under the preceding two rules, the activity materials on the last day of their validity are displayed first. Activities that meet this condition do not need to follow the "activity online time" rule.

• Descriptions about the display of activity materials:

- When the number of advertising spaces in the booth is greater than or equal to the number of materials to be displayed in the booth, all materials are sorted and displayed by combining activity priority, activity freshness, and material freshness.
- When the number of advertising spaces in the booth is less than the number of materials to be displayed in the booth, all materials are sorted by combining activity priority, activity freshness, and material freshness. And the first N materials of which the number is equal to the number of advertising spaces in the current booth are sorted and displaced.

For example, at present, there is a booth containing a three-frame banner advertising space, which is called by Activity A, Activity B, Activity C, and Activity D at the same time, and these activities need to show the advertisement to the same user. Each activity has two materials for display. The priority of Activity A and Activity B is set to high, the priority of Activity C is set to medium, and the priority of Activity D is set to low. That is, the four materials of Activities A and B currently have priority to be displayed. Also, Activity A goes online earlier than Activity B, and Activity A goes offline one day earlier than Activity B. Then, when Activity A is online and not on the last day of its validity, the three-frame advertising space will be allocated to Material 2 and Material 1 of Activity B, and Material 2 of Activity A in turn. When Activity A is on the last day of its validity, the three-frame advertising space will be allocated to Material 2 and Material 1 of Activity A, and Material 2 of Activity B in turn.

In this example, the fatigue rule hasn't taken into account. Combined with the fatigue rule that is observed for each material, the materials of Activity C and Activity D also have the opportunity to be exposed to the same user, which means that the current booth can be used more effectively.

6. (Optional) Click the **+ Add Booth** button again and refer to the preceding steps to add a new booth. Then, you can upload one or more sets of advertising ideas.

# Message push

Only interactive marketing activities support the message push function, and only one message push can be added for an interactive marketing activity.

#### ? Note

To use message push function, the Mobile Push Service component is required for the client. Therefore, you need to initialize the push SDK and bind the user ID and the token before using the function.

Complete the following operations to add a message push:

- 1. In the **Push message** area, click the **+ Add push** button. Then, an edit card of **Information Push Content** will pop up on the page.
- 2. Click the **+ Add material** button to add materials pushed by the activity. Up to three sets of push materials can be added. A message preview is provided on the right side of the Push content card, which refreshs in real time according to the added materials. In a message push, you can add multiple materials. For details on the configuration information for each material, see the following table.

Parameter	Description
Push title	The title of the message. A maximum of 30 characters is allowed.



Push text	The content of the message. A maximum of 100 characters is allowed.
Redirection URL	The redirected link associated with the material. When a user clicks the push text on the client, the user will be immediately redirected to the page that the URL links to. Redirections to online pages and app pages are supported. For different types of pages, you need to configure different information:
Redirection ORL	<ul> <li>Online page: Enter the URL of the target page.</li> </ul>
	<ul> <li>App page: Enter the schema of the target page and perform relevant configurations on the client. For more information, see Access to the client (Access to Android devices, Access to iOS devices, Access to HTML5).</li> </ul>

# 3. Configure **Message validity period** and **No push time period**. The following table shows the parameter descriptions.

Parameter	Description
Message validity period	Configure the validity of message push, with a minimum of three minutes and a maximum of 72 hours. If the device is offline or the user logs off, the message push will fail. Within the message validity, after the device goes online or the user logs on again, the system will push the message again to ensure the delivery rate.
Push Mode	The default mode is <b>Push Now</b> and cannot be modified. After the activity takes effect and goes online, once the marketing event triggered by the user hits the activity, the system will immediately push the material (message) of the activity.
No push time period	Configure the time period that a push does not work. During the set period, no matter whether the current activity is hit or not, the system will not push the material to avoid disturbing the user.

4. After you confirm that the information is correct, click **Submit**.

# 7.4.4. Manage activities

The activity configuration page of an activity unit shows information about all the activities contained in the unit, and you can search for a target activity and view, edit or delete an activity.

# Search for activities

You can search for a target activity in the activity list by activity status, activity type, and activity name/ID keyword. The procedure is as follows:

- 1. In the search box at the upper-right of the activity list, select the activity status, activity type, or enter the activity name or ID as the keyword.
- 2. Click **Search**. Then the system will filter out the activities that meet all the criteria and display the corresponding activity cards.

# View the activity list

In the activity unit list on the **Activity management** page, click the name of the activity unit to enter the activity configuration page of the activity unit.

Various activity cards are displayed on the activity list, as shown in the preceding figure. The activity cards are sorted in descending order of the creation time, and the activity that was created the latest is displayed in the front. Each activity card shows the following information:

- Activity name
- Marketing type
- Start time
- End time
- Activity status

There are three types of activity status: **To be online**, **Online**, and **Offline**.



The status descriptions are as follows:

- To be online: Indicates that the activity has not been taken online. The following two statuses are included:
  - To be online (Not expired): The activity has not been taken online since creation and the activity has not expired.
  - To be online (Expired): The activity has not been taken online since creation, but the activity has expired.

Activities to be online can be edited, deleted, and taken online. After an activity is deleted, the activity data will still be retained in the system for statistical analysis of the effectiveness of the activity.

- Online: Indicates the activity is online. The following two statuses are included:
  - Online (Not started): The activity is online, but the activity has not started (before the activity start time).
  - Online (Started): The activity has been online, and the activity has started.

According to whether the activity has started or not, the online activities can be divided into online (not started) and online (started). Online activities can be taken offline.

- Offline: Indicates the activity has gone offline. The following two statuses are included:
  - Offline (Not expired): The activity has gone offline, and the activity has not expired.

• Offline (Expired): The activity has gone offline, and the activity has expired.

According to whether the activity has launched or not, the offline activities can be divided into offline (not expired) and online (expired). Offline activities cannot be taken online, regardless of whether they have expired or not.

# View activity details

In the activity list, each activity card displays part of the activity information. You can view the full information in the activity details page.

Click **View** in the target activity card to view the details of the activity.

# **Edit an activity**

You can edit the activity information. The procedure is as follows:

NoteActivities in To be online status can be edited.

- 1. Click **Edit** in the target activity card.
- 2. To modify the activity information, see Create an activity for instructions on the procedure and parameters.

# Take an activity online/offline

## Take an activity online

A created activity does not take effect automatically, and you need to manually take it online. Click **Online** in the target activity card to activate it.

#### ? Note

Activities in **To be online** status can be taken online.

- Activities not expired: In the pop-up window, click OK.
- Activities expired: In the pop-up window, click **OK** to enter the **Modify activity** page, modify the validity period of the activity, and take the activity online.

#### ? Note

For expired activities, you need to change the validity period of the activities so that the activities are within the validity period and then take the activities online.

### Take an activity offline

For the activity that are online within the validity period, you can manually take it offline. Click **Offline** in the target activity card to deactivate it.

#### ? Note

Activities in **Online** status can be taken online. The system will automatically deactivate the online activities when the activity end time arrives.

### **Delete an activity**

You can delete an activity. The procedure is as follows:

#### ? Note

Activities in **To be online** status can be deleted. After an activity is deleted, the activity data will still be retained in the system for statistical analysis of the effectiveness of the activity.

- 1. Click ... in the target activity card. In the pop-up window, select **Delete**.
- 2. In the pop-up window, click **OK**.

# 7.4.5. Manage activity units

The management operations of an activity unit include searching, editing, and deleting the activity unit, as well as viewing the activity effect under the activity unit.

Activity manage	ment					
Create an activity unit						Activity unit name/ID Q
Activity unit name	Activity unit ID	Validity period	Creator	Creation time	Activity status	Operation
Test	1993		mpaas_public_test	2021-09-08 12:	52:57 Online 0 To be	e online 0 Offline ( Edit Delete
20200814	1419	2020-08-14 16:44:25 to 2021-09-30 14:32:55	mpaas_public_test	2020-08-14 16:4	44:00 Online 2 To be	online 0 Offline C Effect

# Search for an activity unit

You can search for the target activity unit in the list according to the name or the ID keyword of the activity unit. The procedure is as follows:

- 1. Enter the name or the ID of the activity unit in the search box on the top right of the list.
- 2. Click the search icon, and the result will display in the list.

# Edit an activity unit

#### ? Note

When there are online activities in the activity unit, the activity unit cannot be edited, and no **Edit** operation entry exists in the corresponding **Operation** column.

You can edit the information of an activity unit, including the name of the activity unit and the conversion indicator. The operation method is as follows:

- 1. In the list, click Edit of the Operation column corresponding to the target activity unit.
- 2. In the pop-up dialog box, modify the information of the activity unit. For more information about the parameter descriptions, see Create an activity unit .
- 3. Click the **Confirm** button to confirm the modification.

# Delete an activity unit

#### ? Note

When there are online activities in the activity unit, the activity unit cannot be deleted, and no **Delete** operation entry exists in the corresponding **Operation** column.

The operation method of deleting an activity unit is as follows:

1. In the list, click **Delete** of the **Operation** column corresponding to the target activity unit.

2. In the pop-up dialog box, click **Confirm** to confirm the deletion.

# View the activity effect

In the list of activity units, click **Effect** of the **Operation** column corresponding to the target activity unit. And then you can view the activity effect of the activity unit from multiple dimensions. You can view the activity effect of all activitys or an activity in an activity unit by selecting activitys. When you select and view an activity, you can also view the activity effect of all marketing channels or a marketing channel under the activity.

The information of the activity effect displayed on the activity effect dashboard is divided into **Activity overview** and **Activity retention**. The following is a description of the statistical indicators displayed in each module.

# Activity overview

Activity Overview provides information on data overview, activity effect conversion, and activity effect.

- **Data overview**: The following indicator data all provide day-over-day information. That is the ratio of the change in the data of the currently selected date to the data of the previous day.
  - **Reach count**: The sum of the number of exposures of activity advertisements and the number of successful pushes within a certain period of time.
  - **Deduplicated reach devices**: The sum of the total number of unique devices of any advertising material in a marketing activity and the number of devices successfully receiving pushes within a certain period of time.
  - **Response count**: The sum of the number of advertisement taps and the number of push taps within a certain period of time.
  - **Deduplicated response devices**: The sum of the number of users who tap advertisements and the total number of unique devices on which pushes are tapped within a certain period of time.
  - **Activity share count**: The number of times that the activity is tapped and shared on the activity landing page within a certain period of time. The specific implementation depends on the accuracy of the event tracking on the client, which requires the activity page to support sharing and have the corresponding event tracking.
  - Deduplicated share devices: The total number of unique devices on which the Share button is tapped on the activity landing page within a certain period of time. The specific implementation depends on the accuracy of the event tracking on the client, which requires the activity page to support sharing and have the corresponding event tracking.
  - **Critical event occurrences**: The number of MAS events associated with the activity that occur after users tap the activity advertisements within a certain period of time.
  - **Devices with critical event occurrence**: The total number of unique devices on which the MAS events associated with the activity occur after users tap the activity advertisements within a certain period of time.
- Activity effect conversion: The following indicator data can be viewed from both PV and UV dimensions.
  - **Response rate**: The ratio of the number of responding devices to the number of reached devices.
  - **Share rate**: The ratio of the number of devices with sharing operations to the number of reached devices.
  - **Event conversion rate**: The ratio of the number of devices on which the MAS events associated with the activity occur to the number of responding devices.
- Activity effect: Trends in reach, responses, shares, and critical events after the activity goes online, which can be viewed from both PV and UV dimensions.

User Guide-Console operation





# **Activity retention**

Activity retention provides the retention information of activity users after the activity is delivered. MCDP allows you to view daily, 15-day, and 30-day retention data within one week after delivery. By default, the activity retention date displays the total days before the activity ends, up to 30 days.

vity retenti	on 🛈						2020-06-17	~ 2020-0	8-26
ate	1 day later	2 days later 3 days	s later 4 days	later 5 days later	r 6 days later	7 days later	15 days later	30 da	ays later
06-24	-		-	-	-	-	-	-	-
06-25	-		-	-	-	-		-	-
06-26				-	-	-		-	-
06-27	-		-	-	-	-	-	-	-
06-28	45.13%	2.89%	2.89%	28.69%	-	14.37%			-
06-29	1.93%	1.88%	18.57%	-	9.30%	11.18%	3.72%		-
06-30	27.97%	100.00%		100.00%	100.00%	26.96%	100.00%	-	-

# 7.5. Push management

You can use Push management to configure the fatigue of push requirements for global marketing activities in the App.

The push fatigue refers to the upper limit of the total number of pushes to a single user or device in unit time. When the total number of pushes to a single user reaches the configured upper limit in unit time, the application will stop the push to this user or device. When entering the next unit time, the application will resume the push.

**Note** If a message has not been pushed in the previous unit time and exceeds the required validity period for unpushed messages after entering the next unit time, the message will not be pushed.

# Procedures

Log on to the mPaaS console, select the target application, and perform the following steps to manage the push:

- 1. In the left navigation bar, select **Mobile Content Delivery Platform > Resource management > Push management**.
- 2. Click the **Edit** button in the upper-right corner of the page to enter the edit mode.
- 3. Configure **Push fatigue control**.
  - i. Click **Unit time (Single user)** to pull down the list, and then select the unit time. You can set the unit time as hour, day, or week.
  - ii. Click **Total pushes (Single user)** to pull down the list, and then select the upper limit of the total number of pushes in unit time. You can select an integer from 1 to 10.
- 4. Click the **Save** button to save the configuration.

# 8.FAQ

This topic classifies and provides answers to frequently asked questions about Mobile Content Delivery Platform (MCDP).

# **Common problems**

# How is MCDP billed?

MCDP is billed based on the number of interface calls from your booths and the traffic that is generated by downloading the resources in your booths.

For MCDP pricing details, see Post-paid mode.

# **Questions about console operations**

# Why am I unable to find the ad in the application after I take an activity online?

Perform the following steps to identify the cause:

- Go to the console and check whether the page path of the dynamic booth or the preset ID of the preset booth that you configure for the activity is correct.
- If you use a dynamic booth for your activity, check whether the specified page path conflicts with that of another booth. A page path specifies the location of a booth. Each page path can be used for only one booth. If you use the same page path for multiple booths, these booths all become invalid.

# I configured an image for a startup booth in the console, but why is the image not displayed when I start the application?

Provided that the configurations of the startup page of your application are correct, the reason why the startup image is not displayed may be that the image size does not match the device model.

To avoid this problem, make sure your configuration meet the following requirements:

- For Android devices, at least one image is required to be of 1080 × 1920 pixels.
- For iOS devices, configure images strictly based on the suggested size. Make sure that at least one image meets the requirement.

# Is redirection to HTML5 offline packages supported?

mPaaS 10.1.68 series for Android and iOS allow you to open offline packages. When you configure the page path of an H5 offline package, make sure to add the additional parameters of the offline package page at the end of the page path.

#### • Protocol specification

mpaas://open?h5AppId=3000000&k1=v1&k2=v2...

In the protocol, the keys for fixed judgment are mpaas , open , h5AppId , where:

- h5AppId refers to the offline package ID.
- Other query parameters are transparently passed to the target page.

#### • Example

- Redirect to the offline package with ID 30000000: mpaas://open?h5AppId=30000000
- Redirect to the specified page of the offline package with ID 3000000 aaa.html : mpaas://open?h5AppId=3000000&url=/aaa.html

# Can the size of a banner booth automatically adapt to that of the image configured for the booth?

Yes. When you create a banner booth, set the booth height to 0 so that the size of the booth can automatically adapt to that of the image configured for the booth.

ash 🛈	Pop-up	Bulletin	banner	List	Float	Custom ①			
Booth	material ty	pe							
	🔽 Static in	mage 🔽	GIF animat	ion 🔽	Lottie a	nimation 🔽	H5 page		
ooth po	sition(The	position of	the preset	booth i	s subject	to the actual	lient setting	)	
) Defa	ult position								
			vhen the h	eight is	not confi	gured on the	lient)		
			vhen the h	eight is	not confi	gured on the	lient)		

# **Questions about Android client access**

# I configured a startup ad, but why is it not displayed on the startup page when I start the application for the first time?

After you configure a startup ad, the ad may not be displayed on the startup page when you start the application for the first time. Generally, the reason is that the client fails to pull the required data.

Call the interface as shown in the following code snippet to check whether the required local data exists. The startup ad is displayed on the startup page only when the required local data exists.

```
public static boolean checkIfSplashPrepared() {
    CdpAdvertisementService serviceByInterface =
LauncherApplicationAgent.getInstance().getMicroApplicationContext().findServiceByInterface
CdpAdvertisementService.class.getName());
    return serviceByInterface.checkIfSplashPrepared();
}
```

# For dynamic booths (custom booths), what requirements must be met for displaying a list of advertisements?

The page must contain the ListView widget.

How do I troubleshoot the problem that no booth is displayed on the client after I configure an activity? Use the logcat filter with Advertisement . Check the pulled booth data for troubleshooting.

# Find the data of the relevant booth based on the booth ID. The following code snippet shows a sample log:

getWrapperCallback onSuccess:[SpaceInfo{spaceCode='20201202', iOSViewId='null', android ViewId='null', h5ViewId='null', appId='', spaceObjectList= [SpaceObjectInfo{objectId='15255', contentType='PIC', contentHeight=1080, crontabList=null, behaviors=[SpaceObjectBehavior [behavior=ALWAYS, showTimes=1, closedB yUser=false, jumpedByUser=false, behaviorUpdateTime=0, hadShowedTimes=0]], widgetId='', content='null', hrefUrl='https://mcdp-hz.oss-cnhangzhou.aliyuncs.com/mcdp/ONEXF2EA312171832default/1606900783534/4b3dcb21535681b162977af38d4e6953.png', shortImgUrl='null', action Url='https://www.baidu.com/', gmtStart=1606900730000, gmtEnd=1612084730000, fqColor='null', bqColor='null', textColor='null', widgetColor='null', priority=1, mrpRu leId='', bizExtInfo={picWidth=1080, gmtModified=1606900803000, picHeight=1920, LAYER TY PE=normal}, timeSensitive=false, clientMinVersion='null', clientMaxVersion='null', logE xtInfo={picWidth=1080, gmtModified=1606900803000, picHeight=1920, groupId=13782, LAYER TYPE=normal}, selfAdapt=false}], location='FULL', height=1080, hasPlaceholder=false, us eCacheFirst=false, width=804, reqRpcTime=1606913000272, multiStyle='BANNER', rotationTime=0, close=false, displayMaxCount=1, modifyTime=0, localRuleList=[], extInfo  $=\{ \} \} ]$ 

- 1. Check whether the pulled booth data matches the configured activity booth based on the booth ID (spaceCode=20201202).
- 2. Check whether the array in spaceObjectList contains data.
- 3. Check whether the value of showTimes is greater than the value of hadShowedTimes .
- 4. If you check all the preceding items and find nothing abnormal, check whether other warn logs exist and perform troubleshooting based on alert information.

### **Questions about iOS client access**

# Why am I unable to find the relevant header file?

Add the following code to the xx-mPaaS-Headers.h file that is generated in the MPaaS directory of your project.

#import <MPPromotion/CDPPromotion.h>

# Why is a dynamic booth not displayed?

Perform the following steps:

- 1. Go to the console and check whether the iOS page path (the class name) that you configured for the booth is correct. If the class name is invalid, modify the class name.
- 2. After confirming that the class name is correct, make sure that the class name inherits DTViewController . Otherwise, the booth is not displayed.