Jarcas Promo Code System

v1.0

Instruction Manual

Introduction

Thank you for purchasing the Jarcas Promo Code System (JPCS)! We designed JPCS in the hopes that it will make the task of creating, managing, and redeeming promo codes for in-app purchases a simple and painless one.

NOTE: We have included an example scene showing how JPCS would work in-game. However, since JPCS requires an external web server, you will still have to go through all the Installation and Setup steps before you can test the example scene.

Installation and Setup

JPCS makes use of an external web server providing PHP and mySQL database functionality. You will have to provide this web server yourself. There are many simple and cheap web hosting plans that provide PHP/mySQL.

Once you have your own web server, here are the steps to setup JPCS:

1. Import the JPCS package
2. Configure JPCS Settings
3. Upload PHP files to your web server
4. Run setup.php

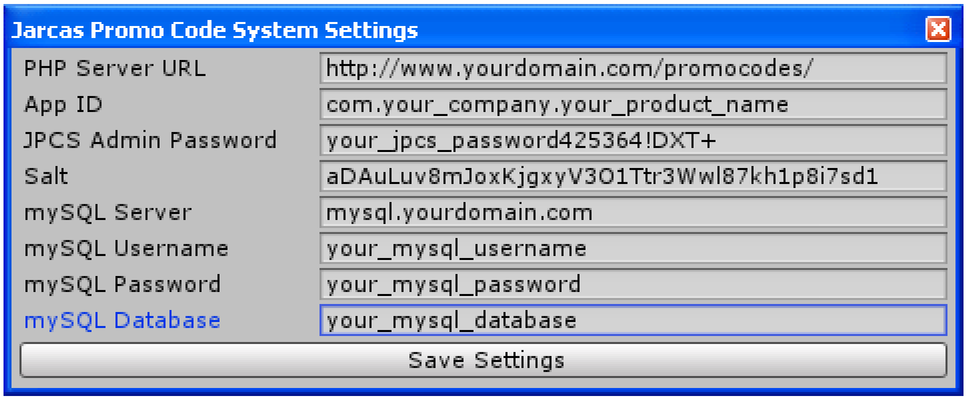
Import JPCS Package

If you haven't already done this, you just need to import the JPCS package from the Asset Store window of the Unity editor.

All of the files will be placed in a directory called *JarcasPromoCodeSystem*. It is highly recommended that you do not move that directory as it may cause problems later if you try to upgrade to a newer version of JPCS.

Configure JPCS Settings

To configure your JPCS Settings, in the Unity editor navigate the menu to *Window->Jarcas Promo Code System->Settings*.



Jarcas Promo Code System Settings

PHP Server URL	http://www.yourdomain.com/promocodes/
App ID	com.your_company.your_product_name
JPCS Admin Password	your_jpcs_password425364!DXT+
Salt	aDAuLuv8mJoxKjgxyV3O1Ttr3Wwl87kh1p8i7sd1
mySQL Server	mysql.yourdomain.com
mySQL Username	your_mysql_username
mySQL Password	your_mysql_password
mySQL Database	your_mysql_database

The available settings are:

- ***PHP Server URL*** - Full URL to the web location where your PHP files will be hosted
- ***App ID*** - An ID that will be associated with all promo codes that you generate. When listing or redeeming codes, only codes associated with this ID will be considered valid.
- ***JPCS Admin Password*** - Admin password for JPCS. Needed to view, generate, or modify promo codes. You should change this from the default value. No need to memorize this password, so a random string of characters (such as those generated by http://passwordsgenerator.net) is ideal.
- ***Salt*** - This is a random string of 40 characters used to secure against hackers spoofing responses from your promo code server. You should change this from the default value. You can generate a random 40 character string at http://passwordsgenerator.net
- ***mySQL Info*** - All the info needed to access the mySQL database on your web server

Some of these settings, such as the JPCS Admin Password and the mySQL login info, are only needed while in the Unity editor so they are only stored locally in *EditorPrefs* and are not packaged into any builds of your project. Other settings which will be required at runtime in-game are stored in the *JPCSRuntimeSettings* ScriptableObject asset.

Also note that when you save your JPCS Settings, a *config.php* file is automatically generated in the *PHPServer* directory. This config file contains the settings necessary for the JPCS server logic to work properly.

Upload PHP Files

Now you will need to upload all the PHP files (including the *config.php* file that was auto-generated when you saved your JPCS Settings) to your server. The PHP files are all in a directory named *PHPServer*. Upload all those files to your server via FTP or whatever method you prefer. Filezilla is a Jarcas Studios favorite for handling FTP transfers.

Make sure that the PHP files are placed in the directory referenced by the *PHP Server URL* field of the JPCS Settings window. Failure to do so will result in 404 HTTP errors in the editor and in game.

Run setup.php

Finally, you'll need to run the setup PHP script on your web server. This script will automatically create the tables in your mySQL database that are necessary for JPCS to operate properly.

All you need to do is open any web browser and go to the URL where your *setup.php* is located (e.g. - *http://www.yourdomain.com/promocodes/setup.php*). When you do this you should see a message indicating that the tables were created successfully and that *setup.php* has been deleted for you.

Congratulations! You're done with setting up JPCS! Now you can get started with integrations or go ahead and create some promo codes using the Promo Code Manager.

Integration

The process of integrating JPCS with your game is very simple. You'll need to do four things:

1. Add the PromoCodeServer prefab
2. Choose some Product IDs
3. Call RedeemCode() from your GUI
4. Write callback functions

Take a look at the *JPCSExample* scene in the *Example* folder if you'd like to see a very simple example of integrating JPCS into a game.

PromoCodeServer Prefab

The PromoCodeServer Prefab is located in the Prefabs folder and must be placed in any scene where you wish to redeem promo codes. The PromoCodeServer component handles all the dirty work of communication with your PHP/mySQL server.

Product IDs

You will need a Product ID for each in-game product you want to be able to redeem with a promo code. If you are planning on implementing In-App Purchases you probably already need to create some Product IDs for those systems. For convenience, you may want to use the same Product IDs for promo codes and your In-App Purchase systems.

Product IDs can be any string of characters you choose. They are used when generating and redeeming promo codes. Each promo code you generate will be stored in your database and associated with exactly one Product ID. When a promo code is redeemed, the server will send a response to your app letting you know what Product ID was redeemed. Based on that Product ID you will award whatever in-game products you desire (coins, level packs, etc.). The implementation of that is entirely up to you.

Call RedeemCode()

Wherever you plan to accept promo codes in your GUI, you just need to add a call like this:

```
PromoCodeServer.instance.RedeemPromoCode( code );
```

`code` is a simple string containing the promo code being redeemed. This call will begin the process of communicating with the server. Depending on the result of the redemption attempt, one of the below callback functions will be called.

Callback Functions

The PromoCodeServer implements two C# events: `OnCodeRedeemSuccess` and `OnCodeRedeemFailure`. Which event is fired depends on whether the code redemption was successful or not. These events are subscribed to with a statement like this:

```
PromoCodeServer.instance.OnCodeRedeemSuccess += YourFunc;
```

You may have multiple subscribers for each event. This may be useful if you want different classes/objects to handle redemption of different Product IDs.

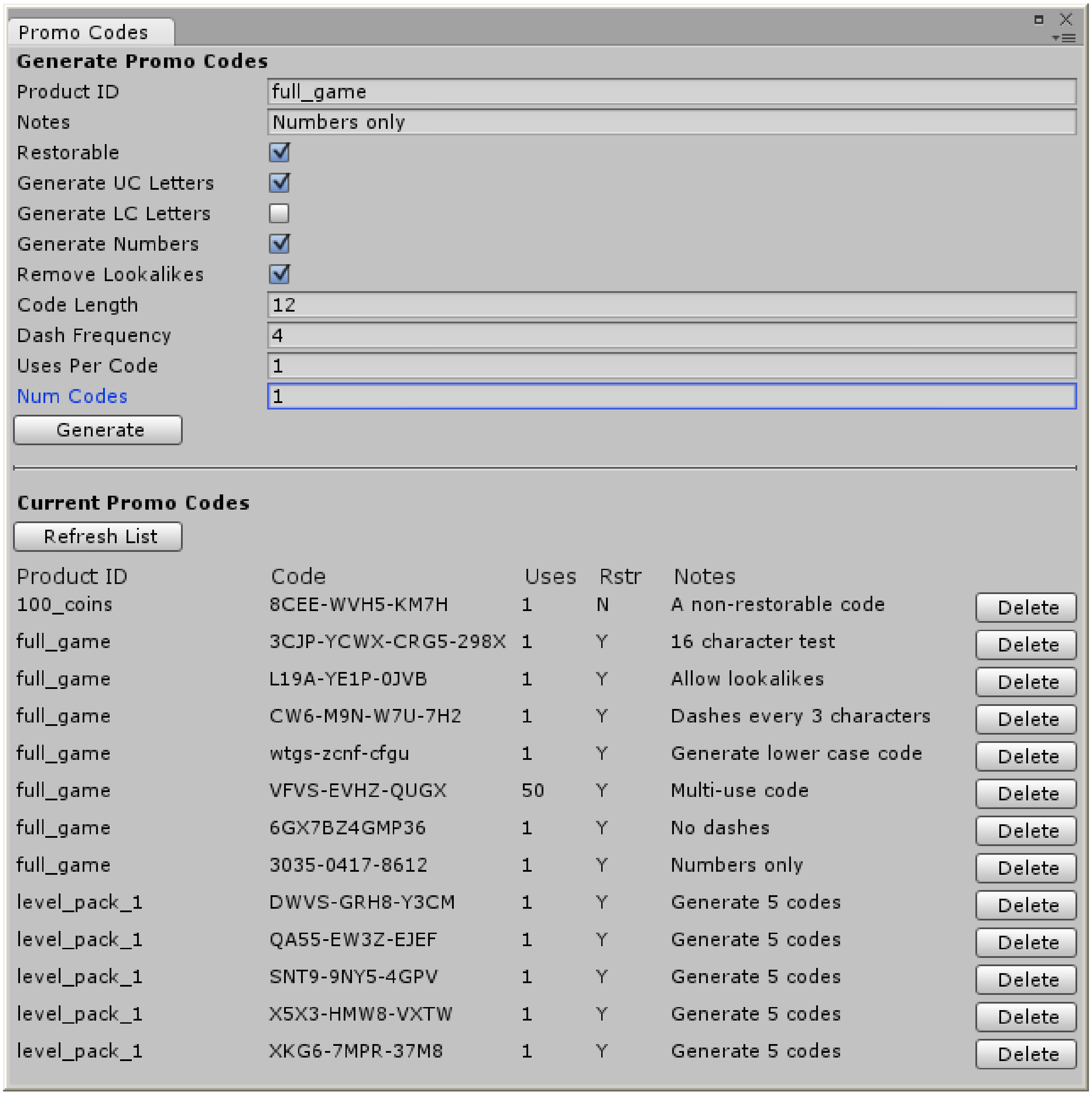
The `OnCodeRedeemSuccess` event is fired with a single string parameter containing the Product ID that was just successfully redeemed. The implementation of what happens based on the Product ID redeemed is entirely up to you.

The `OnCodeRedeemFailure` event is fired with a single string parameter containing the error message of why the code redemption failed.

Managing Promo Codes

NOTE: You must follow the above steps in Installation and Setup prior to managing your promo codes. Failure to do so may result in unexpected behavior.

To manage your promo codes, in the Unity editor navigate the menu to *Window->Jarcas Promo Code System->Manage Promo Codes*.



Generate Promo Codes

Product ID	full_game
Notes	Numbers only
Restorable	☑
Generate UC Letters	☑
Generate LC Letters	□
Generate Numbers	☑
Remove Lookalikes	☑
Code Length	12
Dash Frequency	4
Uses Per Code	1
Num Codes	1

Current Promo Codes

Product ID	Code	Uses	Rstr	Notes	
100_coins	8CEE-WVH5-KM7H	1	N	A non-restorable code	Delete
full_game	3CJP-YCWX-CRG5-298X	1	Y	16 character test	Delete
full_game	L19A-YE1P-0JVB	1	Y	Allow lookalikes	Delete
full_game	CW6-M9N-W7U-7H2	1	Y	Dashes every 3 characters	Delete
full_game	wtgs-zcnf-cfgu	1	Y	Generate lower case code	Delete
full_game	VFVS-EVHZ-QUGX	50	Y	Multi-use code	Delete
full_game	6GX7BZ4GMP36	1	Y	No dashes	Delete
full_game	3035-0417-8612	1	Y	Numbers only	Delete
level_pack_1	DWVS-GRH8-Y3CM	1	Y	Generate 5 codes	Delete
level_pack_1	QA55-EW3Z-EJEF	1	Y	Generate 5 codes	Delete
level_pack_1	SNT9-9NY5-4GPV	1	Y	Generate 5 codes	Delete
level_pack_1	X5X3-HMW8-VXTW	1	Y	Generate 5 codes	Delete
level_pack_1	XKG6-7MPR-37M8	1	Y	Generate 5 codes	Delete

You should now see the Promo Code Manager tab. This tab is dockable so if you want to keep it in your Unity editor layout you can. The top half of the tab handles generation of promo codes. The bottom half handles listing and deleting currently existing promo codes.

Generate Promo Codes

There are several options regarding promo code generation. Some are related to how a promo code can be used. Some are related to the specifics of what characters are used to generate a code. All are described below:

- ***Product ID*** - developer specified ID for use when redeeming a code
- ***Notes*** - notes for use by developer. May show when or to whom the code was given
- ***Restorable*** - A good rule of thumb is to set Restorable to true for any non-consumable products (e.g. - level packs) and false for consumable products (e.g. - in-game currency). Technical details follow. Promo codes are designed such that each code can only be redeemed once by a given device, regardless of how many uses are remaining on the code. When a code is redeemed, the device ID is stored in the server's database as having redeemed that particular code and the number of uses remaining for that code decreases by 1. If that device ID attempts to redeem that same code again, then what happens depends on whether that code is set to Restorable or not. Restorable codes will provide a success response and will not decrease the remaining uses for that code. Non-Restorable codes will provide a failure response that the code has already been used on that device.
- ***Generate UC Letters*** - Should uppercase letters be included in the generated code
- ***Generate LC Letters*** - Should lowercase letters be included in the generated code
- ***Generate Numbers*** - Should numbers be included in the generated code
- ***Remove Lookalikes*** - Should lookalike characters (that may cause confusion when copying or reading codes) be removed from the generated code. Lookalike characters are "O", "0", "1", "L", and "I"
- ***Code Length*** - Length in characters of the generated code
- ***Dash Frequency*** - Should dashes be used as a separator? A value of 0 means don't include any dashes. A value of X means that the code generator should add a dash to the code after every X characters. Note that, regardless, dashes are purely for usability and are ignored when verifying promo codes for redemption.
- ***Uses Per Code*** - Number of times each code can be used. Note that a single device can never redeem a code multiple times (Restorable codes are a special case, see above). So, this value actually determines how many different devices can redeem each code.
- ***Num Codes*** - Number of promo codes to generate

When these codes are generated they are added to the server's database and available for immediate use. The generated codes are also automatically copied to your system clipboard so that you can easily paste them into an e-mail or other document.

Current Promo Codes

You may need to click the *Refresh List* button before you see anything in this section. The list will also automatically refresh any time a new promo code is generated or an existing code is deleted.

This section will list all promo codes found associated with your current App ID in your server's database. Note that although promo codes from multiple apps are stored in the same table in the same database, you will only ever see the promo codes associated with your current App ID in this list.

If you need to change your App ID for your current Unity project you can do that in the JPCS Settings window. Note that changing your project's App ID will not modify existing promo codes. They will all still be associated with their original App ID.

The columns shown in this list are the same as the fields described above in *Generate Promo Codes*. Note that *Rstr* is an abbreviation for Restorable.

You can also delete existing promo codes in this section by pressing the *Delete* button beside each code.

Support and Feedback

Please don't hesitate to contact us at support@jarcas.com if you need any help with JPCS. Feedback is also encouraged. Anything you hate, love, or would like to see added/changed? Please let us know!

Follow Jarcas Studios on Twitter @JarcasStudios

Jarcas Studios
www.jarcas.com