
X-Updater Documentation

Release 0.6.0

StepToSky

Jan 30, 2018

Contents

1	Terms	3
2	For distributors	5
2.1	Howtos	5
2.1.1	How to create new product	5
2.1.2	How to prepare your product	5
2.1.3	How to upload your product or its update	6
2.1.4	How to use ignore list	7
2.2	Dev client (xupd)	7
2.2.1	Installation	7
2.2.2	Commands	7
2.3	Beta/Alpha access	9
2.4	FAQ	10
2.4.1	How to report a bug?	10
2.4.2	How to get support?	10
2.4.3	X-Updater restores files that can be changed by my application. How to tell the X-Updater not to restore them?	11
2.4.4	My product has extended versions. How can I distribute all of them?	11
2.4.5	My product has some variations e.g. different colors, plug-ins, textures etc... How can I distribute those variations?	11
3	For consumers	13
3.1	Java client	13
3.2	FAQ	13
3.2.1	How to report a bug?	13
3.2.2	How to get support?	13
4	Changelog	15
4.1	The X-Updater Server	15
4.1.1	0.7.0 (19.01.2018)	15
4.2	The X-Updater Dev client (xupd)	15
4.2.1	0.2.0 (19.01.2018)	15
4.3	The X-Updater Java Client	16
4.3.1	0.6.1-beta (23.01.2018)	16
4.3.2	0.6.0-beta (19.01.2018)	16
4.3.3	0.5.1-beta (07.07.2017)	16
4.3.4	0.5.0-beta (29.06.2017)	16

4.3.5	0.4.0-beta (01.04.2017)	16
4.3.6	0.3.1-beta (25.12.2016)	16
4.3.7	0.3.0-beta (28.11.2016)	17
4.3.8	0.2.7-beta (28.09.2016)	17
4.3.9	0.2.6-beta (27.09.2016)	17
4.3.10	0.2.5-beta (10.09.2016)	17
4.3.11	0.2.4-beta (24.08.2016)	17
4.3.12	0.2.2-beta (15.08.2016)	17
4.3.13	0.2.0-beta (12.08.2016)	17
4.3.14	0.1.9-beta (07.07.2016)	17

The X-Updater system consists from 4 parts:

- X-Updater web server
- X-Updater client for *distributors*
- X-Updater client for *consumers*
- X-Updater synchronizer with the <http://www.x-plane.org> store

If you wish you can see the repository of this documentation <https://github.com/steposky/x-updater-doc>

Contents:

CHAPTER 1

Terms

administrator The guys who administrate the X-Updater system.

developer The guys who develop the X-Updater system.

distributor A person who uses X-Updater system to distribute updates of his products.

consumer A person who uses X-Updater for getting updates for his products copy.

commit The process for making a new product's snapshot in the system.

This section is for *distributors*

2.1 Howtos

2.1.1 How to create new product

You have to contact the *administrator* he will add your new product into the system and give you instructions.

2.1.2 How to prepare your product

Before you will be able to use the X-Updater for distribution of your products you have to prepare it.

- If you want to setup a new product no server you have to ask the *administrator* to create a new product there.
- Install dev client (named *xupd*) See *Dev client (xupd)*. if you don't have the client you have to ask the *administrator* where you can get it.
- Run `xupd init -product="product id"` See *init*.

Description of the x-updater files:

- *x-updater/client-configuration* - this file is the configuration for the client. Actually you don't have to change it yourself, **but you must commit this file with your product i.e. don't put it into the x-updater ignore list.**
- *x-updater/productId* - this file contains the public id of your product. **It must be committed with your product i.e. don't put it into the x-updater ignore list*.**
- *x-updater/description.txt* - this file should describe your product, for example you can put the changelog there and any other text. **notes:** the content of this file will be shown for the *consumers* while updating. This file is processed while committing so you have to prepare all its content before you *commit* a new snapshot.
- *x-updater/xupignore* - this is the ignore list. See *How to use ignore list*.

Note: The x-updater automatically ignores unnecessary files like *description.txt*, *xupdignore*.

Note:

- The files *info.msg*, *x-updater/info.msg*, *x-updater/x-updater.cnf*, *.xupdignore*, *x-updater/.xupdignore* are not used anymore and should be deleted after `xupd init`.
-

2.1.3 How to upload your product or its update

- Setup your xupd client if you haven't done so. See *client*.
- Put necessary information into `x-updater/description.txt` file.

Warning: If you forget to put changes into this file there is no way to fix it except ask the *administrator*

- Use command `xupd status` if you want to see the changes before uploading them. See *status*.
- Use command `xupd commit` to upload your changes and make new snapshot. Since your changes are uploaded they immediately are available for the *consumers*. See *commit*.

Examples:

```
# It initializes and commits your product
$ xupd init -productId=7b8ea61ebb1082b9e8300c304b182f808d1bccae
$ xupd commit -apiKey=ca5c22db1805612b49264281bffcfa86caa4a879 \
               -productId=7b8ea61ebb1082b9e8300c304b182f808d1bccae \
               -desc="any text here"
```

```
# Sets you distributor api key into local storage.
# It allows you to not specify it each time when you want to use status or commit_
↪command.
$ xupd client set -apiKey=ca5c22db1805612b49264281bffcfa86caa4a879
# go into product folder
$ cd path_to_product
# It initializes your local product in the current working dir
# Also it allows you to not specify productId each time when you want to use status_
↪or commit command.
$ xupd init -productId=7b8ea61ebb1082b9e8300c304b182f808d1bccae
# It checks changes
$ xupd status
# It makes snapshot on the server with your changes
# and starts distributing it
$ xupd commit -desc="any text here"
```

Note: In the previous version after your changes were uploaded you had to run X-Updater-client to initialize preparation process on the server then you had to wait for an email that confirms that the server has indeed prepared the files. Now if `xupd commit` doesn't return an error then all went well.

2.1.4 How to use ignore list

Ignore list is located *x-updater/xupdignore*

Ignore list uses a prefix and postfix conception. The prefix and postfix should be separated by * if there is no * presented then text considered as prefix. For example: *folder2/*ex* any file or folder that starts with *folder2/* and ends with *ex* will be ignored.

Examples:

```
folder1/A - ignore file or whole folder 'A'
folder1/A/* - ignore whole folder 'A'.
folder2/*ex - ignore all files in the folder 'folder2' that end with 'ex'
*ex - ignore all files in all the folders that end with 'ex'
```

You can use *!* or *not* for the files that you don't want to ignore.

This example shows how to ignore whole folder except one file:

```
folder1/A/* - ignore whole folder 'A'.
!folder1/A/myFile - but don't ignore the file 'myFile' in the folder 'A'
# the previous line can also be written the following way:
not folder1/A/myFile
```

Note: All paths must be specified relative to the product folder, not relative to the location of the ignore list!

2.2 Dev client (xupd)

2.2.1 Installation

The client doesn't need special installation just put it somewhere and add to your environment *PATH* the path to the folder where *xupd* is located.

2.2.2 Commands

init

```
$ usage: xupd init [-productId [PRODUCT ID]]
```

Init a new product in you local product's folder. This command will create *x-updater* folder and all necessary files with necessary content. This command doesn't create and prepare product on the server for this issue you have to contact *administrator*.

```
arguments:
    -productId      Product public id.

optional-arguments:
    -dir            Product dir. if it isn't specified then
                   the current working dir is considered as the product's one.
```

Examples:

- Init new product with its public id

```
$ xupd init -productId=7b8ea61ebb1082b9e8300c304b182f808d1bccae
```

client

```
$ usage: xupd client [sub-command] [arguments]
```

This command can be used to manage client settings. The client settings is stored in your PC you can set new values or read current ones.

```
sub-commands
  set          set value
  get          get value

arguments:
  -apiKey      [set, get] distributor's api key
```

Examples:

- Set the *distributor* api key

```
$ xupd client set -apiKey=ca5c22db1805612b49264281bffcfa86caa4a879
```

status

```
$ usage: xupd status [-dir [DIR]] [-apiKey [API KEY]] [-productId [PRODUCT ID]]
```

This command is for checking the difference between your local product and last committed one to the server. It is for information only and it doesn't change anything.

```
optional-arguments:
  -dir          Product dir. if it isn't specified then
                  the current working dir is considered as the product's one.

  -apiKey       Distributor api key, if it isn't specified then
                  the value will be searched in the client settings.

  -productId    Product public id, if it isn't specified then
                  the value will be searched in the file *productId* inside
                  the *x-updater* folder.
```

Examples:

```
$ xupd status
```

```
$ xupd status -dir="myProductFolder" \
  -apiKey=ca5c22db1805612b49264281bffcfa86caa4a879 \
  -productId=7b8ea61ebb1082b9e8300c304b182f808d1bccae
```

commit

```
$ usage: xupd commit [-dir [DIR]] [-apiKey [API KEY]] [-productId [PRODUCT ID]]
                  [-type [TYPE]] [-slots [SLOTS]]
                  [-resetBetaAccess] [-force]
```

This command is for committing and uploading your changes. I.e. this command makes new snapshot in the system.

```
arguments:
    -desc          Commit description, for example you can set version.
                   If you are familiar with git, svn, hg etc...
                   this description is like their one.

optional-arguments:
    -dir           Product dir. if it isn't specified then
                   the current working dir is considered as the product's one.

    -apiKey        Distributor api key, if it isn't specified then
                   the value will be searched in the client settings.

    -productId     Product public id, if it isn't specified then
                   the value will be searched in the file *productId* inside
                   the *x-updater* folder.

    -type          Commit type release, beta, alpha etc...

    -slots         Number of free slots for new public beta consumers.

    -resetBetaAccess Remove all public beta consumers.

    -force         If there is previous unfinished commit you
                   can abort it and force to make this one.
                   This parameter is recommended for continuous integration.
```

Note: See [Beta/Alpha access](#) for more information about *-type*, *-slots*, *-resetBetaAccess*.

Examples:

```
$ xupd commit -desc="any text here"
```

```
$ xupd commit -dir="myProductFolder" \
    -apiKey=ca5c22db1805612b49264281bffcfa86caa4a879 \
    -productId=7b8ea61ebb1082b9e8300c304b182f808d1bccae \
    -desc="any text here" \
    -type=beta \
    -slots=100 \
    -resetBetaAccess \
    -force
```

2.3 Beta/Alpha access

The updater gives you 3 types of snapshots *release*, *beta*, *alpha* with different access level. When you create a new snapshot using the commit command you are given a chose as per the type of the snapshot. X-Updater client will give

the *consumer* will be presented a chose as per which type of snapshot to update.

- release - the widest access type. Any *consumer* who has a valid login and password (key) can download the snapshot.
- beta - this access type has limited access. You can grant the access in three ways:
 - to everyone - which means every *consumer* will have a chose of using the beta snapshot
 - to a selected group - similar to private beta test, only the people you chose will be able to get the beta
 - to a limited number of people (slots) - you select a number of available beta test slots which are filled when a *consumer* selects the beta snapshot download. The slots are used up until the amount of available slots is zero. Then you need to give out more slots. If a slot has been taken by a *consumer* it remains his until you reset the access for all testers.
- alpha - this snapshot can be accessed only by the selected group of beta testers (see above). Thus, any beta tester in the selected group has the chose of getting the alpha, provided you have created a separate alpha version of course. This should be used for quick tests of fixes or new functionality.

Note: The access types are prioritized. This means that if a *consumer* has requested an alpha but your last snapshot was a beta, he will get the beta. The prioritization is as follows: release > beta > alpha.

Examples

- we make a new beta snapshot and add 100 new beta slots. Note, the `-slots` argument can be only used with `-type=beta`

```
$ xupd client commit -desc="my new commit" -type=beta -slots=100
```

- we make a release and reset the beta access for all *consumers* with slots

```
$ xupd client commit -desc="my new commit" -type=release -resetBetaAccess
```

Note: To change the number of available slots, take access away, reset the beta access etc. you can contact the *administrator*. This might be needed if you made a mistake, need to change something or want to add slots without making a new snapshot.

2.4 FAQ

2.4.1 How to report a bug?

You have to contact the *administrator* via email.

Write subject .. code-block:: text

[X-Updater][bug] Subject

2.4.2 How to get support?

You have to contact the *administrator* via email

Write subject .. code-block:: text

[X-Updater][support] Subject

2.4.3 X-Updater restores files that can be changed by my application. How to tell the X-Updater not to restore them?

This problem often occurs if you use the same settings file to configure your application and to store user's data.

In general you have to design your application in such a way that application's data and user's data use different files and you shouldn't distribute the files that are used to store user's data.

Another way is to not distribute settings file and check whether settings file exist by your application. If it doesn't then create it by your application with necessary default values.

But if you can't use the abovementioned ways for some reasons you can use tags in ignore list but it is just a workaround and is not recommended. To enable this you have to contact the *administrator*

2.4.4 My product has extended versions. How can I distribute all of them?

In this context 'extended' is considered a completely new product in x-plane.org store. You have to contact the *administrator* to get the instructions.

2.4.5 My product has some variations e.g. different colors, plug-ins, textures etc. . . How can I distribute those variations?

Actually there is no way to distribute products with variations now. But we know about this problem and we are thinking about how to implement this feature.

Pay attention it might be like 'extended' feature described above but actually it is a different thing and it doesn't depend on the store.

3.1 Java client

For using the x-updater system you should have the *X-Updater-Client.jar* file in the root folder of your plane. If you don't have it you can download the file here <http://update.x-plane.org/> and place it into your plane folder. Then, if you don't have Java 8 runtime installed you have to install it from www.java.com, and run X-Updater-Client.jar, use your e-mail address from x-plane.org and license-key for your product.

If you can't run the *X-Updater-Client.jar* file you should search `how to run jar` in Google.

Warning: The x-updater client works only with java 8 now.

Warning: The x-updater system works only with the keys from x-plane.org

Note: You can also download the latest version of your product directly from the store

3.2 FAQ

3.2.1 How to report a bug?

Create a ticket here <http://flightfactor.aero/ticket>

3.2.2 How to get support?

Create a ticket here <http://flightfactor.aero/ticket>

4.1 The X-Updater Server

4.1.1 0.7.0 (19.01.2018)

- Feature: beta/alpha snapshots.
- Feature: new API.
- Feature: refactoring internal code.
- Feature: Utility xupd for delivering your updates to the server.
- Feature: [help for distributors](#).

Note: Breaking changes

- Now it has other access conception
 - Now it has other conception for uploading your update
-

4.2 The X-Updater Dev client (xupd)

4.2.1 0.2.0 (19.01.2018)

- Feature: release/beta/alpha snapshots.
- Feature: beta slots number.
- Feature: beta access reset.
- Feature: force commit.

- Feature: client's settings.
- Feature: init product.
- Improved: information printing.

4.3 The X-Updater Java Client

4.3.1 0.6.1-beta (23.01.2018)

- Fix: running under mac, the client couldn't be started because of hanging self-update.

4.3.2 0.6.0-beta (19.01.2018)

- Feature: selecting snapshot types release/beta/alpha.
- Feature: fresh installation.
- **Patch: when you select the product dir it now uses the x-updater config from that dir.** It fixes some problems when client is used as installer.
- Fix: the problem with incorrect updating when path contains more than one folder named 'Aircraft'.
- Fix: the problem when some files can be rewritten while updating but they should be untouched.
- Fix: the problem when some files must be downloaded or updated but for some reason it did not work.

4.3.3 0.5.1-beta (07.07.2017)

- Fix: bound products path.

4.3.4 0.5.0-beta (29.06.2017)

- Feature: supporting bound products inside the base product.
- Fix: some minor bugs.

4.3.5 0.4.0-beta (01.04.2017)

- Feature: related products.
- Feature: bound products.
- Patch: using case sensitive path on windows.
- Fix: some minor bugs.

4.3.6 0.3.1-beta (25.12.2016)

- Patch: config can have "cache=0" parameter for disabling the cache.

4.3.7 0.3.0-beta (28.11.2016)

- Patch: some UI elements have been increased.
- Patch: auto-trim for the authentication text (i.e. user name and key).
- Patch: text when product's developer doesn't provide product description.
- Patch: logging.
- Fix: changing mode installer->updater when the client is run inside product's directory.

4.3.8 0.2.7-beta (28.09.2016)

- Feature: count of the files for deleting.
- Feature: auto open finish page after installation is completed.
- Feature: auto open finish page after verification is completed and there no files to update.
- Fix: auto-run the client after self-update. WARNING: It will have the effect in next time.
- Fix: progress bar value when process is completed.
- Fix: floating error after pressing cancel button.
- Fix: some bugs in the config.

4.3.9 0.2.6-beta (27.09.2016)

- Feature: link to the support (on finish page and while critical errors)
- Patch: log improvements.

4.3.10 0.2.5-beta (10.09.2016)

- Patch: internal changes.

4.3.11 0.2.4-beta (24.08.2016)

- Fix: fixes texts.

4.3.12 0.2.2-beta (15.08.2016)

- Patch: internal improvements.

4.3.13 0.2.0-beta (12.08.2016)

- Fix: 60 minutes before timeout now.

4.3.14 0.1.9-beta (07.07.2016)

- Fix: mixed updater's and installer's configurations.

A

administrator, [3](#)

C

commit, [3](#)

consumer, [3](#)

D

developer, [3](#)

distributor, [3](#)