

Peanut Quickstart in 10mins

Get started

[Peanut User Guide v1.0.0](#)

[Peanut - A unified NT3 package solution](#)

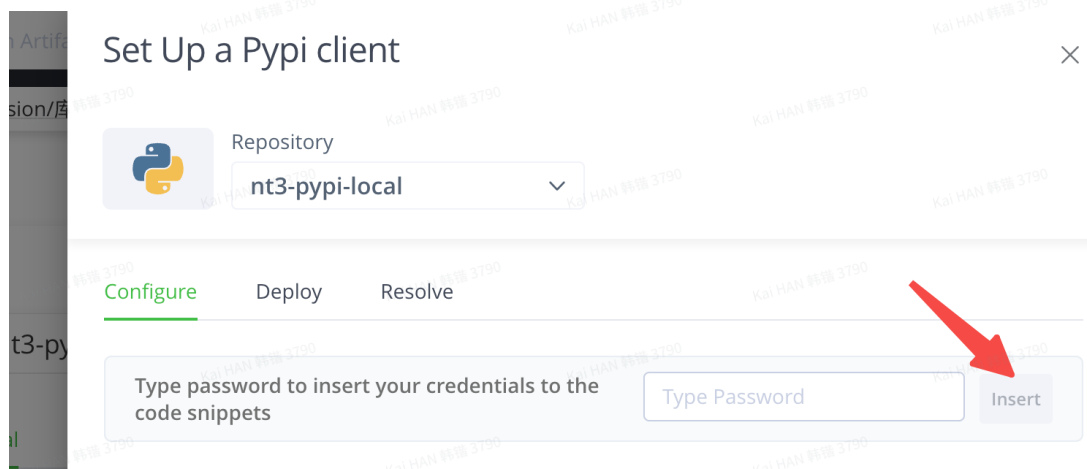
Install peanut&Update peanut

打开仓库页面：

<https://artifactory-cn.nevint.com/ui/repos/tree/General/nt3-pypi-virtual>

Or <https://artifactory-cn.nevint.com/ui/repos/tree/General/nt3-pypi-local>

点击Set Me Up按钮，在Configure Tab中输入密码，点击insert



点击Resolve Tab，获取到用户名和密码，并替换到下面命令里。按说明新建文件~/.pip/pip.conf后，运行如下命令安装

```
1 python3 -m pip install -i https://<UserName>:<Password>@artifactory-cn.nevint.co
```

如需指定安装或更新peanut的版本号，命令行中需制定，例如：`peanut-ntxpkg==1.1.0`，1.1.0为peanut的版本号


! Username和Password可以使用下面的账号，

- Username: dpd-readonly-jenkins-sa:

- Password:
AKCp8k7uem75AP5JbmKde2NbBZ6DmDimbCg7HZDQZskK7oNmtx8SkCxrCqQcDy3
2dKNqeaVgu

或者使用如下的pip.config，这样可以不需要pip install时添加“-i”选项了。



 pip.conf

peanut ntxpkg.yml

下面示例了一个包含在服务源码目录下的ntxpkg的元数据文件，更多的package类型的yml类型可以在peanut的demo工程中看。更详细的资料见：[NTxPKG Specification V1.2.0 \(Draft\)](#)

```
1 name: LoBeamMgr
2 type: component
3 version: 1.23.456
4 prerelease: rc.1
5 group: default-group
6 os:
7   - nvos
8   - Ubuntu
9 arch:
10   - x86
11   - aarch64
12 deployDependencies:
```

```
13 - soa-framework-dev >= 1.3.1
14 - soa-framework >= 1.3.1
15 - peanut-uat:soa-framework == 1.3.1;os=rt-linux;arch=aarch64
16
```

peanut demo

下面的peanut-demo仓库里包含了不同的ntxpkg的场景，大家可以clone下来看看，并真实操作下。

更多demo和文件的介绍：[📖 Peanut Quickstart_Additional](#)

```
git clone git@git.nevint.com:nt3-devops/peanut-demo.git
```



```
1 # 以os为例
2 cd peanut-demo/packages/rt-linux/;
3 # 可以看下os类型的ntxpkg.yml格式
4 cat ntxpkg.yml;
5 # 生成.ntxpkg文件，因为类型为os，所以不用CLI再指定os，但arch需要指定，因为yml文件中有两
6 peanut create --arch x84_64;
7 # .ntxpkg文件已经生成，名称格式：{name}-{version}-{prerelease}.{os}.{arch}.ntxpkg
8 ll rt-linux-1.0.1-rc.1.rt-linux.x84_64.ntxpkg;
9 # 将.ntxpkg文件推送到jfrog上，如果推送失败，可以加上--force，会覆盖之前的.ntxpkg文件
10 peanut publish rt-linux-1.0.1-rc.1.rt-linux.x84_64.ntxpkg
```

```
11 # peanut还有其他参数，比如指定上传到哪个仓库，是否需要强制覆盖等，具体可见文档下方的peanu
12 peanut publish rt-linux-1.0.1-rc.1.rt-linux.x84_64.ntxpkg <--force> <--artifactory>
13
```

peanut commands

具体的命令行说明详见 `peanut --help`，二级命令同样可以加`--help`，例如 `peanut create --help`

```
1 peanut create --help
2 INFO:root:db_dir /Users/kai.han/.ntxpkg/installed_db.yml
3 Usage: peanut create [OPTIONS]
4
5     create ntxpkg
6
7 Options:
8   --target TEXT          target directory
9   --idl TEXT             idl directory
10  --config TEXT          package config file
11  --hooks TEXT           the folder including hook files
12  --os TEXT              OS id
13  --arch TEXT            Arch name
14  --version TEXT         Package version of generated package
15  --prerelease TEXT      prerelease info of generated package
16  --extra TEXT           package extra information, support multiple
17  --revisions TEXT       git revisions, support multiple
18  --no-dep-check         check dependenceis when creating package
19  --git                  create git type package
20  --schema               include package.json to .ntxpkg
21  --group TEXT           Group name of the generated package.
22  --dest TEXT            Group name of the generated package.
23  --include-src TEXT     specify the source included, supports regex,
24                          e.g. dir/.py, support multiple
25  --exclude-src TEXT     specify the source excluded, supports regex,
26  *                       e.g. dir/*.py, support multiple
27  --artifactory-instance TEXT the name of the Artifactory instance in the
28                          config.yml.
29  -h, --help             Show this message and exit.
```

1. peanut create

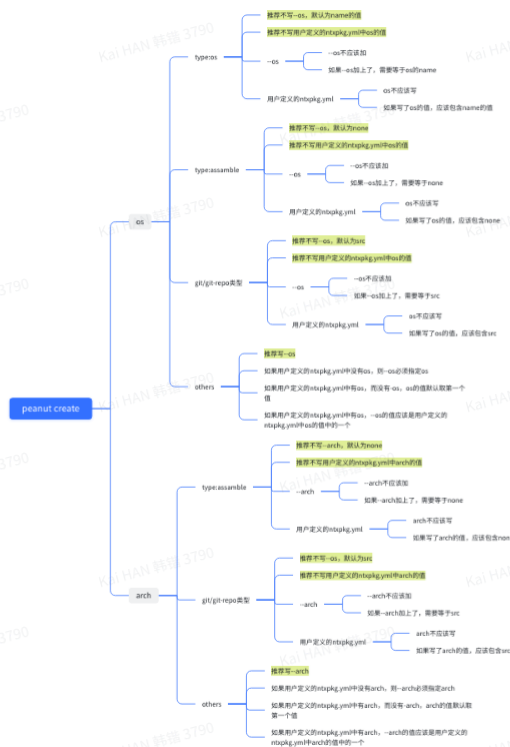
创建ntxpkg格式的包，针对四种不同类型os/component/composite/assemble的包有不同的属性值，还支持git/git-repo类型的格式。具体详情见：

```

1 # common commands
2 peanut create --os rt-linux --arch aarch64 --group default-group
3
4 peanut create --git
5
6 peanut create --revisions git@git.nevint.com:nt3-pet/peanut.git:905d0b61bba24ce7
7
8 peanut create --os rt-linux --arch aarch64 --extra key1=value1 --extra key2=valu

```

peanut create的os和assamble的值规则说明：



2. peanut publish

将create出来的ntxpkg包推送到jfrog对应的包地址中，支持指定不同的artifactory-instance，具体的仓库地址详情与规范详见本文下方：[Peanut Quickstart_Additional](#)

```

1 # common commands
2 # ntxpkg need in current directory, or absolutely path
3 peanut publish vdf-m-1.0.1-rc.1.nvos.tc399.ntxpkg
4
5 # publish ntxpkg to another artifactory
6 peanut publish vdf-m-1.0.1-rc.1.nvos.tc399.ntxpkg --artifactory-instance release
7

```

```
8
9 # want to overwrite the exist ntxpkg
10 peanut publish vdf-m-1.0.1-rc.1.nvos.tc399.ntxpkg --force
```

3. peanut assemble

assemble命令用于将composite package依赖的os package和component package在本地进行组装，会执行如下操作：

- 下载os包，运行mount.sh挂载os target目录下的镜像，
- 依次安装component
- 运行umount.sh卸载镜像

assembly包可以将os, component, composite, assembly package合成一个大的package

```
1 # common commands
2
3 peanut assemble vdf-m==1.0.1-rc.1 --os none --arch none --group default-group
4
5 # assemble ntxpkg from another artifactory
6 peanut assemble vdf-m==1.0.1-rc.1 --os none --arch none --artifactory-instance r
7
```

4. peanut fetch

publish到artifactory的package可以使用fetch命令下载到本地，包括deployDependencies中依赖的ntxpkg包

```
1 # common commands
2 # fetch ntxpkg from jfrog to local
3 peanut fetch vdf-m==1.0.1-rc.1 --os nvos --arch tc399 --group default-group
4 peanut fetch vdf-m==1.0.1-rc.1 --os nvos --arch tc399 --group default-group --de
5
6 # fetch ntxpkg from another artifactory
7 peanut fetch vdf-m==1.0.1-rc.1 --os nvos --arch tc399 --artifactory-instance rel
```

5. peanut source

Component除了普通的Component, 还有两种特殊类型，git component 和 git repo component.

git component 将git repository的revision信息记录在component的ntxpkg.yml中，对其运行source命令可以check out git repository的代码。

```
1 peanut source sdk-core==1.0.1-rc.1 --os src --arch src --group peanut-test-group
```

6. peanut unpack

publish到artifactory的package可以使用unpack命令下载到本地并解压

```
1 peanut unpack test-ecu-sa==1.1.0-rc.1 --os=test-os-sa --arch=x86_64 --dest ~/tes
```

JFrog Instances <artifactory-instance>



The feature is available in v1.1.0 and above. Please install/upgrade the right versions, and remove the original `~/ntxpkg/config.yml`.

The `~/ntxpkg/config.yml` file lists 4 JForg instances - the bleeding-edge, release-candidate, npt-release and vft-release ([国平台/整车制品管理规范 V1.0](#)).

```
1 cache:
2   path: ~/.ntxpkg/cache
3 artifactories:
4   default:
5     host: "https://artifactory-nt3.nioint.com/"
6     url: "https://artifactory-nt3.nioint.com/artifactory/"
7     repo: "devops-bleeding-edge-generic-local/"
8     package-prefix: "artifactory/"
9     tree-prefix: "ui/repos/tree/General/"
10    token: "cmVmdGtuOjAxOjE3MTg0MjI2NDM6VU93UzNSZmtXdUIxN0dxa0hlRlA0e1lpWktK"
11  release-candidate:
12    host: "https://artifactory-nt3.nioint.com/"
13    url: "https://artifactory-nt3.nioint.com/artifactory/"
14    repo: "devops-release-candidate-local/"
15    package-prefix: "artifactory/"
16    tree-prefix: "ui/repos/tree/General/"
17    token: "'[Contact wanjie.gu@nio.com for token]'"
18  npt-release:
19    host: "https://artifactory-nt3.nioint.com/"
20    url: "https://artifactory-nt3.nioint.com/artifactory/"
21    repo: "devops-npt-release-local/"
```

```

22 package-prefix: "artifactory/"
23 tree-prefix: "ui/repos/tree/General/"
24 token: "[Contact wanjie.gu@nio.com for token]"
25 vft-release:
26   host: "https://artifactory-nt3.nioint.com/"
27   url: "https://artifactory-nt3.nioint.com/artifactory/"
28   repo: "devops-vft-release-local/"
29   package-prefix: "artifactory/"
30   tree-prefix: "ui/repos/tree/General/"
31   token: "[Contact wanjie.gu@nio.com for token]"
32 original:
33   host: "https://artifactory-cn.nevint.com/"
34   url: "https://artifactory-cn.nevint.com/artifactory/"
35   repo: "nt3-devops-generic-local/build-system/ntxpkg/"
36   package-prefix: "artifactory/"
37   tree-prefix: "ui/repos/tree/General/"
38   token: "cmVmdGtuOjAxOjE3MDk3MzM4NTk6MkZoZHdBVWZleG1tRXp4NjJwaDVWUm80ajBW"
39 sam:
40   ...


```

To choose different JForg instances, using "--artifactory-instance" option, which is available to all peanut sub-commands, for example,

```

1 The below command will publish package to default (bleeding-edge) instance,
2 > peanut publish service-1.1.1-linux-x86.ntxpkg
3
4 On other hand, the below command will publish package to release-candidate inst
5 > peanut publish service-1.1.1-linux-x86.ntxpkg --artifactory-instance release-

```

To make arifactory-instances other than 'default' work, the peanut user must apply token from JForg admin ( Wanjie GU 顾万杰).

More docs

[📖 Peanut - A unified NT3 package solution](#)