# 模型管理新规范

[**模型包文件名和目录规范**](http://wiki.zte.com.cn/pages/viewpage.action?pageId=32307369)

[**运行期目录规范**](http://wiki.zte.com.cn/pages/viewpage.action?pageId=32307420)

[**版本号规范**](http://wiki.zte.com.cn/pages/viewpage.action?pageId=32099763)

**变化：**

1. **目录**
2. 运行时目录第一层是vnftype，第二层是modelversion。

但模型包中没有这两层，需要在模型包导入时根据version.xml里的属性值自动创建。

1. 原来版本号下直接是cm，fm，现在多加了一层 ：common等。模型包中有多个cm。
2. cm下面又细分yang，lua，json 等。我们不关注，不管cm下的目录怎么分都要能解析。

-vlte //网元类型，version.xml中的 vnftype 属性

----V16.04.01 //模型版本号，version.xml中的modelversion属性

-------version.xml

-------common

----------cm

-------------yang

-------------lua

-------------json

----------fm  
----------pm  
-------[nfcname] //nf的[nfcname]，由网元项目提供，例如sctp，vrouter等  
----------cm

-------------yang

-------------lua

-------------json

----------pm  
----------fm  
----------ddm

-------mt

-------plugins

1. **XML描述文件**

名称变为 version.xml

内容：

1. 节点、属性名全小写。
2. 根节点是model。（原来是NE）
3. vnftype：设备类型（vlte, 5g, …）。

接口中原来的deviceType，改成这个。

查询接口中要用起来，否则不能唯一区分版本。

界面已经显示这一层，但接口变量名有变化。

1. modelversion：模型版本号。

与网元版本号不是一回事。要从网元的配置数据获取模型版本号，根据模型版本号来匹配模型版本。

问题：现在模型里有这个字段了没有？

1. build：build版本号。不一定是补丁号。处理逻辑见“导入模型包”章节。
2. releasetime：发布时间。

查询接口的返回消息中要有这个信息。告警据此判断最新模型。

打补丁时要更新此字段。界面不显示。

1. displayname：界面显示版本号，如果有此字段则显示此字段，此字段为空则显示modelversion
2. desc\_en/desc\_zh：中英文描述。
3. patchinfo： 补丁描述。处理逻辑见“导入模型包”章节。

  <?xml version="1.0" encoding="UTF-8" ?>

<model vnftype="**vlte**"

modelversion="**V16.04.01**"

build="**0002**"

releasetime="**20161201103012**"

displayname="**ITRAN(V1.00.01)**"

desc\_en="**Description of this model**"

desc\_zh="**本模型的中文描述**">

<patchinfo>

  <patch name="**V16.04.01B0002**"

desc\_en="**Description of this patch**"

desc\_zh="**本补丁的中文描述**" />

</ patchinfo>

</ model>

1. **全量的补丁**

现阶段，补丁都是全量的（包含基础版本所有文件，而不仅仅是变化的文件）。

主要原因是增量补丁的CI流程尚未建立，不能保证手工制作的增量补丁都满足合法性校验。

以后可能还是要支持增量补丁。

代码处理时，不管全量还是增量都一样处理。全量就当成一个超大的增量。

1. **导入模型包**
2. **首次导入**
3. build号：

第一次导入的模型包，build版本号就可能不为0。

<model vnftype="**vlte**" modelversion="**V16.04.01**" build="**0002**"/>

Build版本号需要在界面上有显示。在主版本号后面加括号，显示“V16.04.01(B0002)”。

场景：特别是实验室，每个迭代一更新，显示出来更好看。

1. patchinfo：

可能第一次导入的就带有patchinfo。这种情况，直接忽略，不认为是个补丁。

将xml中的patchinfo删掉？避免与后续真正的补丁信息混淆。

1. **后续导入**
2. build号：

界面显示最新build号

1. patchinfo：

如果有patchinfo，则正常记录，界面显示为补丁，可以删除（与现在的实现相同）。

如果没有patchinfo，也应该认为这是一个补丁，需要自动构建一条patchinfo合并到version.xml中。 Patch name 构建为 主版本+build版本，描述置空。举例：

<patchinfo>

<patch name="**V16.04.01B0004**" desc\_en="" desc\_zh=""/>

</patchinfo>

问题：删除的时候如何知道该恢复哪个备份？

1. releasetime：

需要更新

1. **模型更新通知消息**

当前是全量补丁，导入补丁时要有办法比较出变化的目录，到四级目录（cm，fm）。

具体方式待讨论，可选：

模型制作时给每个四级目录生成md5码，放在文件中，导入后保存起来。当导入补丁时判断补丁中各目录的md5码与当前系统中各目录的md5码是否一致，不一致则说明有变化。

发通知时MQ消息中带上变化的目录，也到四级目录。便于各模块判断自己是否关注。