

状态:	转测试	开始日期:	2025-04-28
优先级:	普通	计划完成日期:	
指派给:	马 党旗	% 完成:	0%
类别:		预期时间:	0.00 小时
目标版本:		耗时:	0.00 小时
描述			
网管版本YZMM2.1.0Pre3T1核心网软件管理测试，核心网升级时需要校验版本			
备注：在停止核心网NGC和IMS服务前，需要进行版本校验			

历史记录

- #1 - 2025-04-28 15:58 - 马自伟
- 跟踪从 错误 变更为 功能
 - 状态从 新建 变更为 进行中
 - 指派给从 马自伟 变更为 马党旗

新增功能需求：核心网升级前，判断当前版本和升级版本，是否符合“低升高”规则，如果不符合，则退出升级流程，不再继续

新增版本字符判断逻辑：

```
function extractVersion(versionString) {
  // 转换为小写来忽略大小写差异
  const match = versionString.toLowerCase().match(/rel_(\d+)\.(\d+)\.(\d+)_pre(\d+)/);
  if (match) {
    return {
      major: parseInt(match1, 10), // 主版本
      minor: parseInt(match2, 10), // 次版本
      patch: parseInt(match3, 10), // 修订版本
      pre: parseInt(match4, 10), // 预发布号
    };
  }
  return null; // 无效的版本格式
}

function compareVersions(version1, version2) {
  const v1 = extractVersion(version1);
  const v2 = extractVersion(version2);

  if (!v1 || !v2) {
    return -1;
  }

  // 比较主版本号
  if (v1.major > v2.major) return 1;
  if (v1.major < v2.major) return -1;

  // 比较次版本号
  if (v1.minor > v2.minor) return 1;
  if (v1.minor < v2.minor) return -1;

  // 比较修订版本号
  if (v1.patch > v2.patch) return 1;
  if (v1.patch < v2.patch) return -1;

  // 比较预发布号
  if (v1.pre > v2.pre) return 1;
  if (v1.pre < v2.pre) return -1;

  return 0; // 版本完全相同
}

升级流程中调用：
if (compareVersions(version, targetVersion) < 0) {
```

```
parent.showMessage({message: CURRENT_VERSION + NOT_SUPPORTED + UPGRADE, type: 'error', duration: -1});  
refresh();  
return  
}
```

#2 - 2025-04-29 18:26 - 马党旗

- 状态从进行中变更为转测试