



金蝶 BOS 子系统树指引

文档版本	V1.0
发布日期	2011-5-3

金蝶软件（中国）有限公司

地址	深圳市高新技术产业园南区科技南十二路2号金蝶软件园 Kingdee Software（China）Co.,Ltd
邮编	518057
网址	http://www.kingdee.com http://dev.kingdee.com
客户服务电话	86-755-26612299
客户服务传真	86-755-26615016
客户服务邮箱	webmaster@kingdee.com

版权所有 © 金蝶软件（中国）有限公司

本书著作权属于金蝶软件(中国)有限公司所有，在未经本公司许可的情况下，任何单位或个人不得以任何方式对本书的部分或全部内容擅自进行增删，改编，节录，翻译，翻印，改写。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 子系统树概述	5
2 子系统树数据组成	6
3 子系统树存放的位置和相关的配置文件	6
4 子系统树的生成及加载	9
4.1 子系统树的生成	9
4.2 子系统树的加载	11
5 子系统树的特殊维护	12
5.1 改变节点上下级	12
5.2 新增子系统树	12
6 常见问题分析处理	12
6.1 子系统树问题的排查步骤	12
6.2 通用方法	15
6.3 典型子系统树问题案例	16

1 子系统树概述

BOS 平台中存在多个业务功能相对独立的子系统，如权限、预警、单据转换、编码规则、工作流等。每个子系统都会根据自身的业务，管理辖下的所有相关的业务单元，子系统树即是管理、展示该业务子系统的目录结构的对象。

特点

存在多个子系统树。子系统树具有数据量大，生成过程持续时间长，占用资源多，生成后相对稳定。

类型

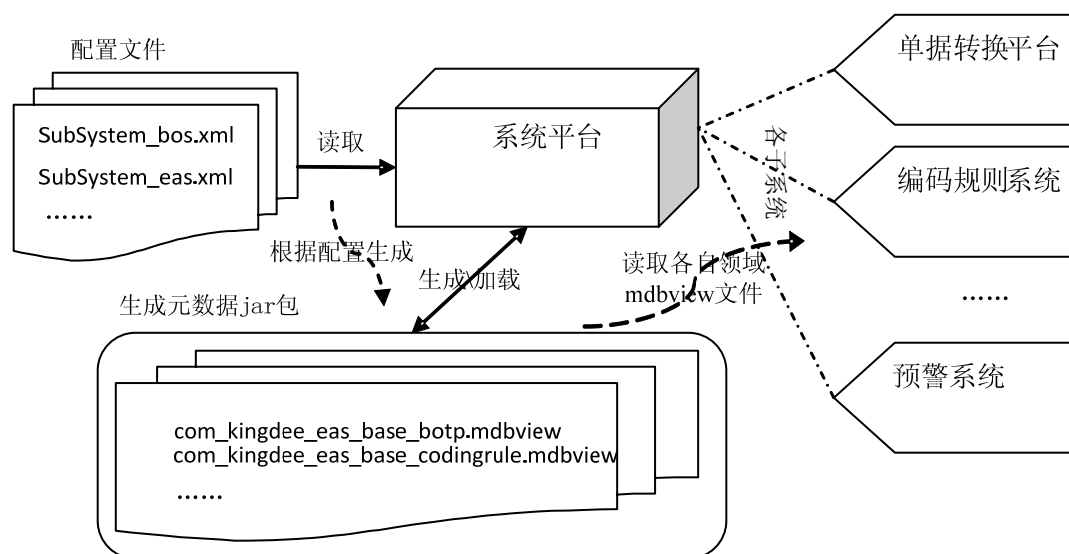
主要存在如下子系统树：参数树、BOTP 树、编码规则树、字段权限树、预警树、查询树、子系统实体树、缓存树、包树、子系统树、对帐中心树。

子系统整体介绍

如下图实示，系统平台在生成子系统树的时候，是根据子系统树相关的配置文件，如 subsystem_bos.xml、subsystem_eas.xml 等，生成各个子系统所需要的目录树文件*.mdbview。

在运行期，加载已经生成的各种*.mdbview 文件，提供个各个子系统使用。

图1-1 子系统树体系简图



2 子系统树数据组成

表2-1 子系统树数据组成表

子系统树类型	数据组成
参数树	由元数据包组成。 参数树在参数管理、参数设置中使用。在“参数管理”中出现的是完整的参数树，在“参数设置”中出现的子系统树,是将没有参数的节点过滤后的参数树。
BOTP 树	由元数据包，实体组成。 实体必须继承于 CoreBillBase，并且没有设置扩展属性 nobotp。
预警树	由元数据包，实体，功能组成。
编码规则树	由元数据包，实体组成。 实体必须设置了扩展属性，codingRuleEnabled,其值为 true;
字段权限树	由元数据包，实体组成。实体必须继承于 ObjectBase，并且设置扩展属性 enableFieldPermission，其值为 true。
预警树	由元数据包，实体，FAÇADE 组成。
查询树	由元数据包，关联查询 QUERY 组成。
子系统实体树	由元数据包，实体组成。
缓存树	由元数据包，实体组成。 实体必须设置了扩展属性，isConfigCache,其值为 true。
包树	由元数据包组成。
子系统树	由元数据包，实体，功能，数据表，界面，关联查询，业务功能组成。

3 子系统树存放的位置和相关的配置文件

标准产品，默认子系统树存储在元数据文件 mdbview-metas.jar 中。

如果有二次开发，通过 bim 生成子系统树，发布和部署，会存储在 sp-bim-metas.jar 或 sp-bim-metas-update.jar 中

如果有做过子系统树的客户化定制，定制的内容会存储在 Server\server\profiles\server1\config\subsystem 的文件夹下

相关文件如下表所示。

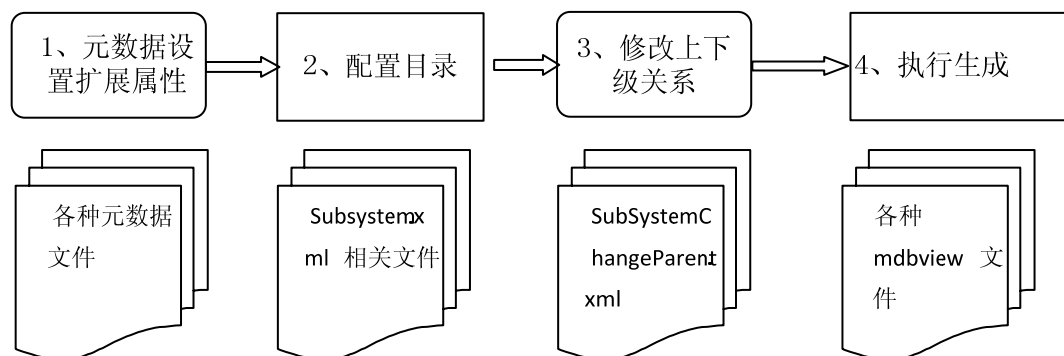
表3-1 相关文件表

文件	位置	用途
SubSystem_bos.xml SubSystem_eas.xml SubSystem_industry.xml SubSystem.xml(V60 解耦之前版本使用)	服务端： Eas\server\properties	配置当前系统所有元数据目录路径。 生成子系统树的时候，根据此配置文件中的目录进行搜索遍历路径下的所有元数据。
SubSystemChangeParent_bos.xml SubSystemChangeParent_eas.xml SubSystemChangeParent_industry.xml SubSystemChangeParent.xml(V60 解耦之前版本使用)	服务端： Eas\server\properties	调整子系统树节点的上下级
SubSystemFilter.xml	服务端： eas\server\properties	新增子系统树使用
mdbview-metas.jar	服务端： Eas\server\metas\bos(服务端加载使用) Eas\server\deploy\fileserver.ear\ easWebClient\metas\bos(提供给客户端下载) 客户端： eas\client\metas\bos	默认标准产品的所有子系统树文件存储的包。 部署二次开发，安装补丁，重新生成子系统树，会更新此包内容。

sp-bim-metas.jar	服务端： Eas\server\metas(服务端加载使用) Eas\server\deploy\fileserver.ear\ easWebClient\metas(提供给客户端下载) 客户端： eas\client\metas	二次开发所有的元数据。 如果在部署到服务器之前，在 bos 工具中生成过子系统树，则此二次开发包根目录下会包含有二次开发相关的子系统树文件*.mdbview，那么在服务端重新生成子系统树之后，此二次开发包中的*.mdbview 文件会被更新。相同内容的*.mdbview 也会更新到 mdbview-metas.jar 中。如果 sp-bim-metas.jar 本身不包含 *.mdbview，那么服务端重新生成子系统树，也不会更新到此包中。
sp-bim-metas_updater.jar	• 服务端： Eas\server\deploy\fileserver.ear\ easWebClient\metas • 客户端： • eas\client\metas	此包是作为增量部署所使用，在某些 EAS 版本中会看到。 其内容有时为空，有时包含 *.mdbview。对于包含有子系统树文件的此包来说，需要注意其中的文件是否根据重新生成而更新(通过查看其中的*.mdbview 文件的时间来判断)。

4 子系统树的生成及加载

4.1 子系统树的生成



步骤 1 根据需要对相关元数据设置扩展属性

对于某些子系统树需要对元数据的扩展属性进行过滤，例如 codingrule 的子系统树，需要配置 codingruleenable 的扩展属性；botp 的子系统树，需要配置 nobotp 的扩展属性。（详细说明请参考标准产品的子系统树说明描述。）

步骤 2 将对应的元数据所在的目录加入到子系统树目录配置文件中。

系统生成子系统树的过程中，会根据“子系统树目录配置文件”中配置的路径（默认包括 app、client 目录），遍历其下的元数据，但不会遍历其他的子目录。

子系统树目录配置文件：

- BOS V6.0 解耦之前版本：
 - 服务端环境：eas\server\properties\SubSystem.xml
 - 开发环境：解决方案根目录下\SubSystem.xml
- BOS V6.0 解耦之后版本：
 - 服务端环境：
 - eas\server\properties\SubSystem_bos.xml(二次开发使用)
 - eas\server\properties\SubSystem_eas.xml(标准产品使用)
 - eas\server\properties\SubSystem_industry.xml(行业产品使用)
 - 开发环境：在解决方案根目录下\ SubSystem_bos.xml、SubSystem_eas.xml、SubSystem_industry.xml

对于标准产品 SubSystem_eas.xml、和行业产品 SubSystem_industry.xml，这个 2 个配置文件，一般不需要现场维护，产品出厂设置即可。

对于 SubSystem_bos.xml，如果现场有二次开发的内容，特别是开发的元数据所在的路径是新建的，务必检查确认 SubSystem_bos.xml 文件中已经包含对应的二次开发的元数据所在的路径。

步骤 3 是否要修改节点的上下级关系

可以通过修改 SubSystemChangeParent.xml 文件来更改默认元数据的上下级关系。满足某些子系统树节点在显示的时候，调整目录树位置的需要。

- BOS V6.0 解耦之前版本：
 - 服务端环境：eas\server\properties\SubSystemChangeParent.xml
 - 开发环境：解决方案根目录下\SubSystemChangeParent.xml
- BOS V6.0 解耦之后版本：
 - 服务端环境：
eas\server\properties\SubSystemChangeParent_bos.xml(二次开发使用)
eas\server\properties\ SubSystemChangeParent _eas.xml(标准产品使用)
 - 开发环境：在解决方案根目录下\ SubSystemChangeParent _bos.xml、
SubSystemChangeParent _eas.xml

步骤 4 生成子系统树

生成操作：

- 在服务端主控台上，通过选择“工具-产生子系统树”菜单生成。
主要在以下情况使用：新安装补丁出现异常中断，手工执行生成(安装补丁默认会执行生成子系统树步骤)，部署二次开发的内容。凡涉及到服务端元数据内容更新的，都需要在服务端重新生成子系统树。
- BOS 开发环境
在 BOS 开发环境，通过选择菜单“BOS 建模工具-工具-生成子系统树”，或者“BOS 建模工具-工具-从包更新子系统树”来生成。

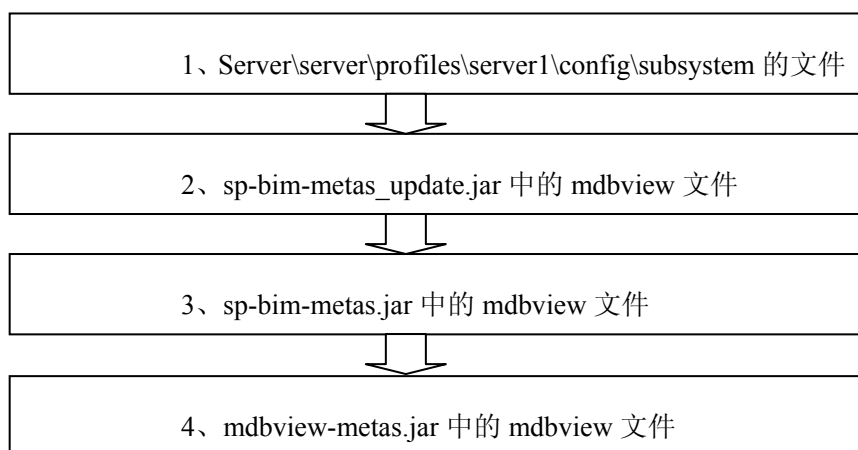
生成文件及路径：

- 服务端主控台生成操作，会生成各种业务领域的 mdbview 为扩展名的子系统树元数据文件，新生成的 mdbview 文件会更新到服务端：eas\server\metas\bos\mdbview-metas.jar，eas\server\deploy\fileserver.ear\easWebClient\metas\bos\mdbview-metas.jar 中；

服务端如果已经部署有包含 mdbview 文件的二次开发元数据包 sp-bim-metas.jar(必须是此名称)，那么服务端重新生成子系统树之后，该二次开发包 sp-bim-metas.jar 中的 mdbview 文件会被更新，如该开发包 sp-bim-metas.jar 中不包含 mdbview 文件，则重新生成的 mdbview 不会更新到此 jar 包中。

- BOS 开发工具生成子系统树，会重新生成解决方案根目录下/metadata 以及 deploy_meta 目录下的所有 mdbview 文件。

4.2 子系统树的加载



上图为子系统树文件的加载优先级顺序，在系统存在多份 mdbview 子系统树文件的情况下，按照上述优先级顺序进行加载。



“Server\server\profiles\server1\config\subsystem”的文件夹下的子系统树一般项目中不会有，主要针对做过子系统树的客户化定制的项目。

5 子系统树的特殊维护

5.1 改变节点上下级

对于需要显示时改变默认的元数据上的上下级关系则需要修改

Server\server\properties\SubSystemChangeParent.xml 文件，此文件一项配置如下

```
<changeParent fullName="com.kingdee.eas.cm.bb.app.DeptIndirectFeeWork"
newParent="com.kingdee.eas.cm.dp" />
```

其中 fullName 表示一个需要改变位置的元数据，newParent 为改变后的位置。

5.2 新增子系统树

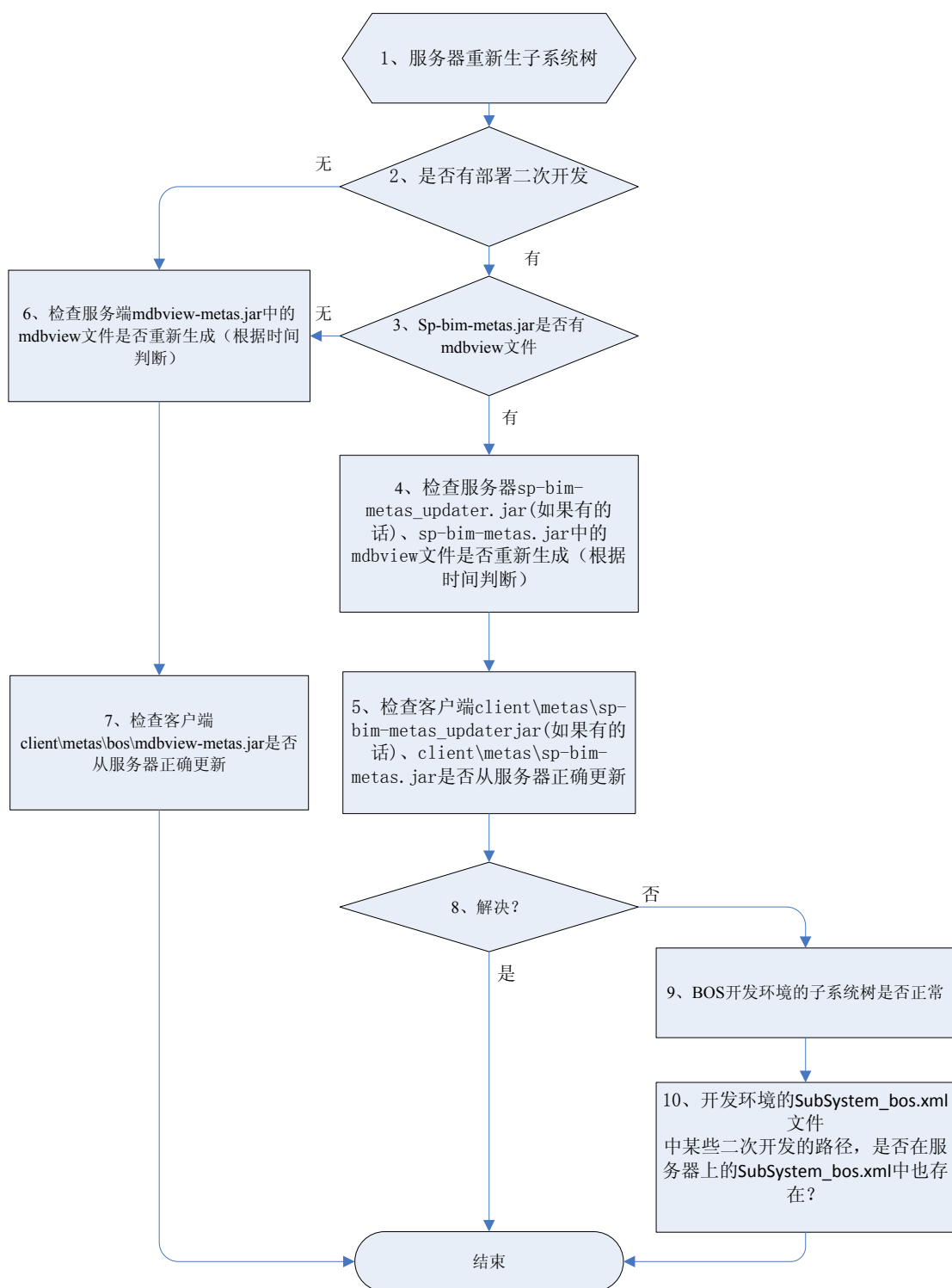
如果要新增子系统树，需要修改 Server\server\properties\SubSystemFilter.xml 文件，具体可以参考该配置文件已有的相关子系统树节点。

```
<!-- 例子
<subSystemFilter>
  <name>com_kingdee_eas_base_botp</name>
  <metaDataTypes>
    <metaDataType>enty</metaDataType>
  </metaDataTypes>
  <extendedPropertys>
    <extendedProperty>
      <type>enty</type>
      <name>nobotp</name>
      <value>>false</value>
    </extendedProperty>
  </extendedPropertys>
  <entityChain>com.kingdee.eas.framework.app.CoreBillBase</entityChain> <package></package>
</subSystemFilter> -->
```

6 常见问题分析处理

6.1 子系统树问题的排查步骤

子系统树问题排查步骤如下图所示。



步骤	操作	处理
1	略	略

2	略	略
3	Sp-bim-metas.jar、 sp-bim-metas_updater.jar 包中是否有 mdbview，取决于部署之前，在 bos 工 具中是否有生成过子系统树	略
4	检查服务端 sp-bim-metas_updater.jar、 sp-bim-metas.jar，查看两个包根目录下的 mdbview 文件是否是最新生成子系统 树的时间	如果服务端重新生成过子系统树，那么此 jar 包中的 mdbview 文件时间戳应该是生成子系统树的时 间，如果时间很早，则说明生成子系统树没有正 确更新此包，并查看服务端生成过子系统树程中 是否有异常(查看控制台日志)。
5	检查客户端的 sp-bim-metas_updater.jar、 sp-bim-metas.jar，检查其中的 mdbview 文件是否跟服务器上的一致	不一致，则说明客户端没有正确更新，需要分析 客户端为什么没有正常更新。 或者换另外的客户端进行验证。
6	检查服务端的 mdbview-metas.jar 中的 *.mdbview 文件，查看是否最近重新生 成子系统树的时间	如果时间跟最近生成子系统树的时间不一致，则 mdbview-metas.jar 中的子系统树没有正确更新， 需要重新生成验证，并查看服务端生成过子系统 树程中是否有异常(查看控制台日志)。
7	检查客户端的 mdbview-metas.jar 中的 *.mdbview 文件，检查其中的 mdbview 文件是否更服务器上的一致	不一致则说明客户端没有正确更新，同步骤 5 处 理。
8	如果以上检查之后，所有是的 jar 包都 是正确更新的，那么则需要分析配置文 件	在分析服务端的配置文件之前，需要确认，BOS 开发环境子系统树是否正确。

9	如果项目有二次开发，需要验证是否在 BOS 开发环境中相关子系统树是否正常	如果开发环境的子系统树也不正常，则需要分析：子系统树看不到新开发的内容，则需要检查 SubSystem_bos.xml 中是否存在新开发的元数据所在的包路径，如果没有，手工更新，然后重新生成子系统树验证。 如果标准产品的某些子系统树受到二次开发的影响，请检查 Subsystem_bos.xml 文件中是否存在标准产品(有行业二次开发的情况下，会存在行业相关的目录)的目录路径，如果存在，手工删除，然后重新生成系统树验证。
10	检查 BOS 开发环境的 SubSystem_bos.xml 文件跟服务器的同名文件，对比是否有差异	检查下解决方案根目录下的 SubSystem_bos.xml 文件，跟服务器上的同名文件是否存在差异？ SubSystem_bos.xml 文件一般包含有二次开发的元数据的路径，如果某些项目存在手工部署的情况，那么此配置文件一般难保证二次开发会得同步更新到服务器，需要人工比较之后，手工修改，将二次开发相关的路径更新到服务器上对应的配置文件中。然后重新生成子系统树验证。

以上问题分析处理的核心步骤：检查服务器、客户端的包更新情况，检查子系统树配置文件情况。

6.2 通用方法

子系统树问题大多因为系统有过二次开发导致，针对有二次开发的子系统树问题，此提供一种通用方法，步骤：

步骤 1 检查 subSystem_bos.xml 文件，删除其中标准产品的路径：如里面包含有如下信息，那么就删除该行：<row>，而其中二次开发的路径要保留。

```
<row fullname="com.kingdee.eas.fi" name="fi" alias_l1="" alias_l2="财务系统" alias_l3="財務系統" isleaf="0" parent="com.kingdee.eas" />
```

步骤 2 删除服务器上(2 处)sp-bim-metas.jar 包中的根目录下的所有*.mdbview 文件。

步骤 3 如果 server\deploy\fileserver.ear\casWebClient\metas 下有 sp-bim-metas_updater.jar，则将其剪切到备份路径(或删除)。

步骤 4 在服务端主控台重新生成子系统树。

6.3 典型子系统树问题案例

1) 二次开发的内容在编码规则、单据转换等子系统树目录中看不到

分析处理：

这里是指二次开发的相关内容，在所有子系统树目录中看不到，根据上面的分析步骤，如果开发环境就看不见，那么则可能是开发的元数据目录在没有更新到 SubSystem_bos.xml 文件中。

在 bos 开发工具中，如果是通过“建模工具”新建的组别，在该组别下新建的业务单元，发布的时候，工具是会自动更新当前新建组别到 SubSystem_bos.xml 文件中的，如果是通过“设计开发工具”新建的包，则直接发布该包下新建的元数据，其包路径不会自动更新到 SubSystem_bos.xml 文件中，因此，需要检查确认下 SubSystem_bos.xml 文件中是否存在新建的二次开发的路径，然后再行生成子系统树验证

2) 二次开发的内容在开发环境，各个子系统树正常，部署到服务器之后，服务器端重新生成子系统树，但是看不见二次开发的内容

分析处理：

首先根据上述的分析步骤，检查对应的 jar 包是否正确更新，在正确更新的基础上，检查服务器上的 SubSystem_bos.xml 跟 bos 开发环境下的 SubSystem_bos.xml 是否有差异，是否二次开发的相关路径没有正确更新。手工修复一致后，重新生成子系统树验证。

3) 部署到服务器上的二次开发的内容，各个子系统树里能看见部分二次开发的内容，另外部分的二次开发的内容找不到

分析处理：

结合上述各问题的分析，应该是部分二次开发的内容是通过 bos 设计开发工具直接新建的元数据，且该元数据所在路径也是通过设计开发工具新建，直接在设计开发工具中发布，导致此

路径相关的元数据在相关的子系统树中看不到。需要手工修改 SubSystem_bos.xml，保证相关的路径在此文件中正确配置，再行重新生成子系统树验证。

4) 行业版产品，做了二次开发，部署到服务器，各个子系统树目录结构不正确，丢失子目录、丢失各个节点等现象

分析处理：

根据之前的类似的问题，在 SubSystem_bos.xml、SubSystem_eas.xml、SubSystem_industry.xml 三份子系统树的配置文件中，各 xml 文件中的路径一般不会有重复，某些情况下，通过 bos 工具修改发布标准产品的单据，或者行业的单据，会将单据相关的路径做为二次开发的路径更新到 SubSystem_bos.xml 文件中，而此单据的路径本来在 SubSystem_eas.xml 或者 SubSystem_industry.xml 中就已经存在，生成子系统树的时候，出现问题中描述现象。需要手工删除在 SubSystem_bos.xml 中，和 SubSystem_eas.xml 及 SubSystem_industry.xml 中重复的目录，然后重新生成子系统树验证。

5) 服务端生成子系统树异常

分析处理：

服务端生成子系统树的问题一般两种原因：

- 服务端的目录的读写权限导致生成子系统树异常，需要对 EAS 服务端的安装目录的权限重新设置，特别是 EAS 服务端目录下的 JDK 目录，需要设置可读写的权限。
- 某些项目上，备份服务端的元数据文件，直接保存在系统元数据的文件夹，并且备份的文件命名中包含有中文。

以上情况，检查处理后重新生成并验证。

6) 二次开发部署之后，重新生成了子系统树，“参数管理-财务系统-存货核算”节点丢失

分析：

该问题的项目，对供应链下的某单据进行发布，bos 工具将

```
<row fullname="com.kingdee.eas.scm" name="scm" alias_l1="scm" alias_l2="供应链管理" alias_l3="供应链管理" isleaf="0" parent="com.kingdee.eas"/>
```

```
<row fullname="com.kingdee.eas.scm.cal" name="calculate" alias_l1="calculate" alias_l2="存货核算" alias_l3="存货覈算" isleaf="1" parent="com.kingdee.eas.scm"/>
```

相关的目录追加更新到在 subSystem_box.xml 中，而“存货核算”参数由于之前版本的子系统树配置文件定义的时候有些特殊，导致由工具生成更新到 subSystem_box.xml 中的节点描述，跟标准产品的 subSystem_eas.xml 文件中的对应节点描述不一致，最终使得生成的参数节点子系统树丢失“存货核算”。

处理：

删除 subSystem_box.xml 中的有关供应链“com.kingdee.eas.scm”，以及其子目录的所有目录节点。然后重新生成子系统树。

7) 由于手工部署导致的子系统树问题

分析：

某些项目由于各种原因，会出现二次开发手工部署元数据的问题，并且，部署到服务器的元数据包，命名没有按照规范统一命名为 sp-bim-metas.jar，导致服务器元数据目录下存在各式各样名称的二次开发的元数据包，且有些元数据包中还包含了*.mdbview 文件。

服务端生成子系统树，只会更新 sp-bim-metas.jar 包和 mdbview-metas.jar 包中的*.mdbview。那些没有及时更新的自定义名称的二次开发元数据包中的子系统树，导致了各种各样的子系统树问题。

处理：

删除相关的自定义名称的二次开发包中的*.mdbview 文件。