

开发环境到测试环境发布手册

发布包结构及命名规则

发布包结构应如下:

内容可根据实际情况做适当调整

```
发布文件包
|   ITSM软件资源库模块部署实施V19090601.docx
|
├─发布文件
|   └─应用包
|       |   biz-0.0.1-SNAPSHOT.jar
|       |   bpm-0.0.1-SNAPSHOT.jar
|       |   gateway-0.0.1-SNAPSHOT.jar
|       |   html.zip
|       |   permission-0.0.1-SNAPSHOT.jar
|       |   resources-0.0.1-SNAPSHOT.jar
|       |
|       └─流程
|           |   软件入库申请_版本4.txt
|           |   软件安装申请_版本9.txt
|           |
|       └─数据库脚本
|           └─A01-升级脚本
|               |   A00-数据清理.sql
|               |   A01_create_tables.sql
|               |   A02-datapriv.sql
|               |   A10_ORG.sql
|               |   A11_POSITION.sql
|               |   A20-USERS.SQL
|               |   A30-APP.sql
|               |   A40-ROLES.SQL
|               |   A50-Dic.sql
|               |
|           └─A02-数据字典
|               |   A00_清除数据字典数据.sql
|               |   A10_FRAME_APP_DATA_DICTIONARY.sql
|               |   A20_FRAME_APP_DATA_DICTIONARY_TYPE.sql
|               |
|           └─B01-系统管理
|               |   A00-清理脚本.sql
|               |   A01-FRAME_DATA_PRIV_TYPE.sql
|               |   A02-FRAME_DATA_PRIVILEGE_DEF.sql
|               |   A20-FRAME_RES_PROJECT.sql
|               |   A21-FRAME_RES_MODEL.sql
|               |   A22-FRAME_RES_SUBMODEL.sql
|               |   A23-FRAME_RES_SERVICES.sql
```

- | A24-FRAME_RES_PAGE.sql
- | A25-FRAME_RES_PAGE_FUNC.sql
- | A26-FRAME_RES_FUNC_SERVICE_REL.sql
- | A90-全局通用字段.SQL
- | A91-数据表分类.sql
- |
- ├B02-软件白名单
- | A00-清理脚本.sql
- | A10_FRAME_RES_MODEL.sql
- | A20_FRAME_RES_SUBMODEL.sql
- | A30_FRAME_RES_SERVICES.sql
- | A40_FRAME_RES_PAGE.sql
- | A50_FRAME_RES_PAGE_FUNC.sql
- | A60_FRAME_RES_FUNC_SERVICE_REL.sql
- | A70_FRAME_RES_FUNC_REF_PAGE.sql
- | A80_白名单升级.sql
- |
- ├B03-资源管理
- | A00-清理脚本.sql
- | A01-FRAME_RES_MODEL.sql
- | A02-FRAME_RES_SUBMODEL.sql
- | A03-FRAME_RES_SERVICES.sql
- |
- ├B04-流程管理
- | A00-清理脚本.sql
- | A10-FRAME_RES_MODEL.sql
- | A20-FRAME_RES_SUBMODEL.sql
- | A30-FRAME_RES_SERVICES.sql
- | A40-FRAME_RES_PAGE.sql
- | A50-FRAME_RES_PAGE_FUNC.sql
- | A60-FRAME_RES_FUNC_SERVICE_REL.sql
- | A70-FRAME_RES_FUNC_REF_PAGE.sql
- | A80-BPM升级脚本.sql
- |
- ├B05-公共页面
- | A00-清理脚本.sql
- | A10-FRAME_RES_MODEL.sql
- | A20-FRAME_RES_SUBMODEL.sql
- | A30-FRAME_RES_SERVICES.sql
- | A40-FRAME_RES_PAGE.sql
- | A50-FRAME_RES_PAGE_FUNC.sql
- | A60-FRAME_RES_FUNC_SERVICE_REL.sql
- |
- ├C01-角色管理
- | A00-数据清理.sql
- | A10-FRAME_AUTH_ROLE_PAGE.sql
- | A11-FRAME_AUTH_ROLE_PAGEFUNC.sql
- |
- └C02-APP信息
- | A00-数据清理.sql
- | C0209-FRAME_APP_INFO.sql
- | C0210-FRAME_APP_ORG.sql
- | C0211-FRAME_APP_MENU_LIST.sql

A.岗位(不导)

将以下脚本查找出来的数据导出成insert脚本即可。根据实际情况替换code

```
select *
from FRAME_WORK_POSTION
where CODE_ = '05';
```

岗位线

将以下脚本查找出来的数据导出成insert脚本即可。根据实际情况替换code

```
select *
from FRAME_WORK_POSTION_TYPE
where NAME_ = '测试岗位线';

select *
from FRAME_WORK_POSTION_RELATE
where TYPE_ID_ in (select FRAME_WORK_POSTION_TYPE.OID
                   from FRAME_WORK_POSTION_TYPE
                   where NAME_ = '测试岗位线');
```

B.组织机构维护(不导)

将以下脚本查找出来的数据导出成insert脚本即可。根据实际情况替换code

```
select *
from FRAME_DEPARTMENT_TYPE
where CODE_ = 'jiguan';
```

C.角色(参考数据隔离策略)

将以下脚本查找出来的数据导出成insert脚本即可

```
--清理
delete from FRAME_ROLE_INFO where CODE_ in ('ROLE_YJYWRY', 'ROLE_YWBG-GLY', 'ROLE_YWBG-
JDZHZRR', 'ROLE_YWBG-YDZHZRR', 'ROLE_YWBGLD', 'ROLE_YWBG_PTRY', 'opsreport');
-- 导出
select * from FRAME_ROLE_INFO where CODE_ in ('ROLE_YJYWRY', 'ROLE_YWBG-GLY', 'ROLE_YWBG-
JDZHZRR', 'ROLE_YWBG-YDZHZRR', 'ROLE_YWBGLD', 'ROLE_YWBG_PTRY', 'opsreport');
```

D.数据字典

将以下脚本查找出来的数据导出成insert脚本即可.然后在新环境同步

导出脚本

```
-- 从树形节点导出,替换DEV
--导出类型
SELECT *
FROM FRAME_APP_DATA_DICTIONARY_TYPE T
WHERE T.ORG_ID = '1' AND T.DELETE_STATUS_ = 0
CONNECT BY T.PARENT_ID_ = PRIOR T.OID
START WITH T.OID IN (SELECT OID FROM FRAME_APP_DATA_DICTIONARY_TYPE A WHERE A.CODE_ =
'DEV');

--导出值
SELECT *
FROM FRAME_APP_DATA_DICTIONARY A
WHERE A.DELETE_STATUS_ = 0 AND
A.TYPE_ID_ IN
(SELECT OID
FROM FRAME_APP_DATA_DICTIONARY_TYPE T
WHERE T.ORG_ID = '1' AND T.DELETE_STATUS_ = 0
CONNECT BY T.PARENT_ID_ = PRIOR T.OID
START WITH T.OID IN (SELECT OID FROM FRAME_APP_DATA_DICTIONARY_TYPE A WHERE
A.CODE_ = 'DEV'));
```

同步脚本

```
--数据字典同步及数据修正
--同步类型
INSERT INTO FRAME_APP_DATA_DICTIONARY_TYPE
(OID
, CODE_
, NAME_
, DESP_
, PARENT_ID_
, REMARK_
, ENABLED_
, SEQUENCING_
, ORG_CODE_
, VERSION_
, LOCKED_STATUS_
, DELETE_STATUS_
, CREATE_DATE_
, CREATE_USER_
, UPDATE_DATE_
, UPDATE_USER_
, TENANT_CODE_
```

```

, DATA_SECRET_LEVCODE_
, DATA_ORG_ID_
, DATA_ORG_CODE_
, DATA_ORG_LEVCODE_
, DATA_DEPT_ID_
, DATA_DEPT_CODE_
, DATA_DEPT_LEVCODE_
, LEAF_NODE_
, APP_CODE_
, IS_SYSTEM
, IS_VISIBLABLE
, IS_EDITABLE
, ORG_ID)
SELECT LOWER(SYS_GUID()) AS NOID
, A.CODE_
, A.NAME_
, A.DESP_
, A.PARENT_ID_
, A.REMARK_
, A.ENABLED_
, A.SEQUENCING_
, A1.DEPT_CODE_ ORG_CODE_
, 1 VERSION_
, 0 LOCKED_STATUS_
, 0 DELETE_STATUS_
, A.CREATE_DATE_
, A.CREATE_USER_
, A.UPDATE_DATE_
, A.UPDATE_USER_
, A.TENANT_CODE_
, A.DATA_SECRET_LEVCODE_
, A.DATA_ORG_ID_
, A.DATA_ORG_CODE_
, A.DATA_ORG_LEVCODE_
, A.DATA_DEPT_ID_
, A.DATA_DEPT_CODE_
, A.DATA_DEPT_LEVCODE_
, A.LEAF_NODE_
, A.APP_CODE_
, A.IS_SYSTEM
, A.IS_VISIBLABLE
, A.IS_EDITABLE
, A1.OID ORG_ID
FROM FRAME_APP_DATA_DICTIONARY_TYPE A
JOIN (SELECT *
      FROM FRAME_APP_DATA_DICTIONARY_TYPE T
      WHERE T.ORG_ID = '1' AND T.DELETE_STATUS_ = 0
      CONNECT BY T.PARENT_ID_ = PRIOR T.OID
      START WITH T.OID IN (SELECT OID FROM FRAME_APP_DATA_DICTIONARY_TYPE A WHERE
A.CODE_ = 'DEV')) A0 ON A0.OID =

      A.OID
JOIN (SELECT T.*

```

```

        FROM FRAME_DEPARTMENT_INFO T
        JOIN FRAME_DEPARTMENT_TYPE A ON A.CODE_ = T.TYPE_CODE_
        WHERE A.ORG_TYPE_ = 1 AND A.DELETE_STATUS_ = 0 AND T.DELETE_STATUS_ = 0 AND
T.PARENT_ID_ <> '0') A1 ON 1 = 1;

```

--同步字典

```

INSERT INTO FRAME_APP_DATA_DICTIONARY

```

```

(OID
,NAME_
,CODE_
,VALUE_
,TYPE_ID_
,SEQUENCING_
,REMARK_
,ORG_CODE_
,ENABLED_
,VERSION_
,LOCKED_STATUS_
,DELETE_STATUS_
,CREATE_DATE_
,CREATE_USER_
,UPDATE_DATE_
,UPDATE_USER_
,TENANT_CODE_
,DATA_SECRET_LEVCODE_
,DATA_TENANT_ID_
,DATA_ORG_ID_
,DATA_ORG_CODE_
,DATA_ORG_LEVCODE_
,DATA_DEPT_ID_
,DATA_DEPT_CODE_
,DATA_DEPT_LEVCODE_
,IS_SYSTEM
,IS_VISIBLABLE
,IS_EDITABLE
,ORG_ID)

```

```

SELECT LOWER(SYS_GUID()) OID
, F0.NAME_
, F0.CODE_
, F0.VALUE_
, F1.OID TYPE_ID_
, F0.SEQUENCING_
, F0.REMARK_
, F2.DEPT_CODE_ ORG_CODE_
, F0.ENABLED_
, F0.VERSION_
, F0.LOCKED_STATUS_
, F0.DELETE_STATUS_
, F0.CREATE_DATE_
, F0.CREATE_USER_
, F0.UPDATE_DATE_
, F0.UPDATE_USER_
, F0.TENANT_CODE_

```

```

, F0.DATA_SECRET_LEVCODE_
, F0.DATA_TENANT_ID_
, F0.DATA_ORG_ID_
, F0.DATA_ORG_CODE_
, F0.DATA_ORG_LEVCODE_
, F0.DATA_DEPT_ID_
, F0.DATA_DEPT_CODE_
, F0.DATA_DEPT_LEVCODE_
, F0.IS_SYSTEM
, F0.IS_VISIBLABLE
, F0.IS_EDITABLE
, F2.OID ORG_ID
FROM (SELECT *
      FROM FRAME_APP_DATA_DICTIONARY A
      WHERE A.DELETE_STATUS_ = 0 AND
            A.TYPE_ID_ IN
            (SELECT OID
             FROM FRAME_APP_DATA_DICTIONARY_TYPE T
             WHERE T.ORG_ID = '1' AND T.DELETE_STATUS_ = 0
             CONNECT BY T.PARENT_ID_ = PRIOR T.OID
             START WITH T.OID IN (SELECT OID FROM FRAME_APP_DATA_DICTIONARY_TYPE A
WHERE A.CODE_ = 'DEV')))) F0
JOIN FRAME_APP_DATA_DICTIONARY_TYPE R0 ON R0.OID = F0.TYPE_ID_ AND R0.ORG_ID = '1'
JOIN (SELECT T.*
      FROM FRAME_DEPARTMENT_INFO T
      JOIN FRAME_DEPARTMENT_TYPE A ON A.CODE_ = T.TYPE_CODE_
      WHERE A.ORG_TYPE_ = 1 AND A.DELETE_STATUS_ = 0 AND T.DELETE_STATUS_ = 0 AND
            T.PARENT_ID_ <> '0') F2 ON 1 = 1
JOIN FRAME_APP_DATA_DICTIONARY_TYPE F1 ON F1.CODE_ = R0.CODE_ AND F1.ORG_ID =
F2.ORG_ID;

```

错误时清理脚本

```

--清除字典分类
SELECT *
FROM FRAME_APP_DATA_DICTIONARY_TYPE T
CONNECT BY T.PARENT_ID_ = PRIOR T.OID
START WITH T.OID IN (SELECT OID FROM FRAME_APP_DATA_DICTIONARY_TYPE A WHERE A.CODE_ =
'DEV');

DELETE FROM FRAME_APP_DATA_DICTIONARY_TYPE
WHERE OID IN (SELECT T.OID
              FROM FRAME_APP_DATA_DICTIONARY_TYPE T
              CONNECT BY T.PARENT_ID_ = PRIOR T.OID
              START WITH T.OID IN (SELECT OID FROM FRAME_APP_DATA_DICTIONARY_TYPE A
WHERE A.CODE_ = 'DEV'));

--清理无效的字典值
SELECT *
FROM FRAME_APP_DATA_DICTIONARY T

```

```
WHERE NOT EXISTS (SELECT 1 FROM FRAME_APP_DATA_DICTIONARY_TYPE A WHERE A.OID =  
T.TYPE_ID_);
```

```
DELETE FROM FRAME_APP_DATA_DICTIONARY T  
WHERE NOT EXISTS (SELECT 1 FROM FRAME_APP_DATA_DICTIONARY_TYPE A WHERE A.OID =  
T.TYPE_ID_);
```

E.流程

1. 导出流程定义,流程管理中找到流程,点击 导出发布 按钮.保存附件
2. 在新环境中导入流程,流程管理中左上角"发布配置",上传附件.重新配置环节选人规则

F.页面权限配置

模块

将以下脚本查找出来的数据导出成insert脚本即可.

```
-- 基础数据举例  
SELECT T.* FROM FRAME_RES_MODEL T WHERE T.MODEL_CODE = 'ITSM.基础数据';  
  
SELECT A1.*  
FROM FRAME_RES_SUBMODEL A1  
WHERE EXISTS (SELECT 1 FROM FRAME_RES_MODEL T WHERE T.MODEL_CODE = 'ITSM.基础数据' AND  
T.MODEL_ID = A1.Model_Id);  
  
SELECT T.*  
FROM FRAME_RES_SERVICES T  
WHERE EXISTS (SELECT 1  
FROM FRAME_RES_SUBMODEL A  
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID  
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础数据');  
  
SELECT A1.*  
FROM FRAME_RES_SERVTBL_REL A1  
WHERE A1.SERVICE_ID IN (SELECT T.SERVICE_ID  
FROM FRAME_RES_SERVICES T  
WHERE EXISTS (SELECT 1  
FROM FRAME_RES_SUBMODEL A  
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID  
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE =  
'ITSM.基础数据')));  
  
SELECT A2.*  
FROM FRAME_RES_SERVPRIV_DEFAULT A2  
WHERE A2.SERV_TBL_RELID IN  
(SELECT A1.SERV_TBL_RELID  
FROM FRAME_RES_SERVTBL_REL A1
```



```

WHERE A1.SERVICE_ID IN
(SELECT T.SERVICE_ID
FROM FRAME_RES_SERVICES T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础
数据')));

```

```

SELECT T.*
FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础数据');

```

```

SELECT T1.*
FROM FRAME_RES_PAGE_FUNC T1
WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE =
'ITSM.基础数据'));

```

```

SELECT T2.*
FROM FRAME_RES_FUNC_SERVICE_REL T2
WHERE T2.PAGEFUNC_ID IN
(SELECT T1.PAGEFUNC_ID
FROM FRAME_RES_PAGE_FUNC T1
WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND
T1.MODEL_CODE = 'ITSM.基础数据')));

```

```

SELECT T2.*
FROM FRAME_RES_FUNC_REF_PAGE T2
WHERE T2.PAGEFUNC_ID IN
(SELECT T1.PAGEFUNC_ID
FROM FRAME_RES_PAGE_FUNC T1
WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND
T1.MODEL_CODE = 'ITSM.基础数据')));

```

公共(不导)

将以下脚本查找出来的数据导出成insert脚本即可.

--页面配置信息

```
SELECT T.*
FROM FRAME_RES_PAGE T
JOIN FRAME_RES_SUBMODEL T1 ON T1.SUBMODEL_ID = T.SUBMODEL_ID
WHERE T.PAGE_CODE IN ('GGYM') AND T1.SUBMODEL_CODE = 'ITSM.公共.基础';
```

--页面功能配置信息

```
SELECT *
FROM FRAME_RES_PAGE_FUNC A
WHERE A.PAGE_ID IN (SELECT T.PAGE_ID
                    FROM FRAME_RES_PAGE T
                    JOIN FRAME_RES_SUBMODEL T1 ON T1.SUBMODEL_ID = T.SUBMODEL_ID
                    WHERE T.PAGE_CODE IN ('GGYM') AND T1.SUBMODEL_CODE = 'ITSM.公共.基础');
```

--页面服务关联信息

```
SELECT *
FROM FRAME_RES_FUNC_SERVICE_REL A1
WHERE A1.PAGEFUNC_ID IN
    (SELECT A.PAGEFUNC_ID
     FROM FRAME_RES_PAGE_FUNC A
     WHERE A.PAGE_ID IN (SELECT T.PAGE_ID
                        FROM FRAME_RES_PAGE T
                        JOIN FRAME_RES_SUBMODEL T1 ON T1.SUBMODEL_ID = T.SUBMODEL_ID
                        WHERE T.PAGE_CODE IN ('GGYM') AND T1.SUBMODEL_CODE = 'ITSM.公共.基础'));
```

--页面子页面关联信息

```
SELECT *
FROM FRAME_RES_FUNC_REF_PAGE A2
WHERE A2.PAGEFUNC_ID IN
    (SELECT A.PAGEFUNC_ID
     FROM FRAME_RES_PAGE_FUNC A
     WHERE A.PAGE_ID IN (SELECT T.PAGE_ID
                        FROM FRAME_RES_PAGE T
                        JOIN FRAME_RES_SUBMODEL T1 ON T1.SUBMODEL_ID = T.SUBMODEL_ID
                        WHERE T.PAGE_CODE IN ('GGYM') AND T1.SUBMODEL_CODE = 'ITSM.公共.基础'));
```

--页面所引用的服务信息

```
SELECT *
FROM FRAME_RES_SERVICES A2
WHERE A2.SERVICE_ID IN
    (SELECT A1.SERVICE_ID
     FROM FRAME_RES_FUNC_SERVICE_REL A1
     WHERE A1.PAGEFUNC_ID IN
        (SELECT A.PAGEFUNC_ID
         FROM FRAME_RES_PAGE_FUNC A
         WHERE A.PAGE_ID IN (SELECT T.PAGE_ID
                            FROM FRAME_RES_PAGE T
```

```

JOIN FRAME_RES_SUBMODEL T1 ON T1.SUBMODEL_ID =
T.SUBMODEL_ID
WHERE T.PAGE_CODE IN ('GGYM') AND T1.SUBMODEL_CODE =
'ITSM.公共.基础')));

```

清理页面权限数据

```

--清除页面引用子页面的信息
SELECT *
FROM FRAME_RES_FUNC_REF_PAGE T2
WHERE T2.PAGEFUNC_ID IN
(SELECT T1.PAGEFUNC_ID
FROM FRAME_RES_PAGE_FUNC T1
WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND
T1.MODEL_CODE = 'ITSM.基础数据'))));

DELETE FROM FRAME_RES_FUNC_REF_PAGE T2
WHERE T2.PAGEFUNC_ID IN
(SELECT T1.PAGEFUNC_ID
FROM FRAME_RES_PAGE_FUNC T1
WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND
T1.MODEL_CODE = 'ITSM.基础数据'))));

--清除页面引用服务的信息
SELECT *
FROM FRAME_RES_FUNC_SERVICE_REL T2
WHERE T2.PAGEFUNC_ID IN
(SELECT T1.PAGEFUNC_ID
FROM FRAME_RES_PAGE_FUNC T1
WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND
T1.MODEL_CODE = 'ITSM.基础数据'))));

DELETE FROM FRAME_RES_FUNC_SERVICE_REL T2
WHERE T2.PAGEFUNC_ID IN

```

```

(SELECT T1.PAGEFUNC_ID
 FROM FRAME_RES_PAGE_FUNC T1
 WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
                       FROM FRAME_RES_PAGE T
                       WHERE EXISTS (SELECT 1
                                    FROM FRAME_RES_SUBMODEL A
                                    JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
                                    WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND
T1.MODEL_CODE = 'ITSM.基础数据')));
--清除页面功能信息
SELECT *
FROM FRAME_RES_PAGE_FUNC T1
WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
                     FROM FRAME_RES_PAGE T
                     WHERE EXISTS (SELECT 1
                                  FROM FRAME_RES_SUBMODEL A
                                  JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
                                  WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE =
'ITSM.基础数据'));

DELETE FROM FRAME_RES_PAGE_FUNC T1
WHERE T1.PAGE_ID IN (SELECT T.PAGE_ID
                     FROM FRAME_RES_PAGE T
                     WHERE EXISTS (SELECT 1
                                  FROM FRAME_RES_SUBMODEL A
                                  JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
                                  WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE =
'ITSM.基础数据'));
--清除页面配置信息
SELECT *
FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
              FROM FRAME_RES_SUBMODEL A
              JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
              WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础数据');

DELETE FROM FRAME_RES_PAGE T
WHERE EXISTS (SELECT 1
              FROM FRAME_RES_SUBMODEL A
              JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
              WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础数据');

--清除后台服务数据隔离策略信息
SELECT *
FROM FRAME_RES_SERVPRIV_DEFAULT A2
WHERE A2.SERV_TBL_RELID IN
      (SELECT A1.SERV_TBL_RELID
       FROM FRAME_RES_SERV_TBL_REL A1
       WHERE A1.SERVICE_ID IN
            (SELECT T.SERVICE_ID
             FROM FRAME_RES_SERVICES T
             WHERE EXISTS (SELECT 1
                          FROM FRAME_RES_SUBMODEL A

```

```
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础
数据')));
```

```
DELETE FROM FRAME_RES_SERVPRIV_DEFAULT A2
WHERE A2.SERV_TBL_RELID IN
(SELECT A1.SERV_TBL_RELID
FROM FRAME_RES_SERV_TBL_REL A1
WHERE A1.SERVICE_ID IN
(SELECT T.SERVICE_ID
FROM FRAME_RES_SERVICES T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础
数据')));
```

--清除后台服务关联表信息

```
SELECT *
FROM FRAME_RES_SERV_TBL_REL A1
WHERE A1.SERVICE_ID IN (SELECT T.SERVICE_ID
FROM FRAME_RES_SERVICES T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE =
'ITSM.基础数据')));
DELETE FROM FRAME_RES_SERV_TBL_REL A1
WHERE A1.SERVICE_ID IN (SELECT T.SERVICE_ID
FROM FRAME_RES_SERVICES T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE =
'ITSM.基础数据')));
```

--清除后台服务信息

```
SELECT T.*
FROM FRAME_RES_SERVICES T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础数据');
```

```
DELETE FROM FRAME_RES_SERVICES T
WHERE EXISTS (SELECT 1
FROM FRAME_RES_SUBMODEL A
JOIN FRAME_RES_MODEL T1 ON T1.MODEL_ID = A.MODEL_ID
WHERE A.SUBMODEL_ID = T.SUBMODEL_ID AND T1.MODEL_CODE = 'ITSM.基础数据');
```

--清除子模块信息

```
SELECT *
FROM FRAME_RES_SUBMODEL A1
```

```

WHERE EXISTS (SELECT 1 FROM FRAME_RES_MODEL T WHERE T.MODEL_CODE = 'ITSM.基础数据' AND
T.MODEL_ID = A1.Model_Id);

DELETE FROM FRAME_RES_SUBMODEL A1
WHERE EXISTS (SELECT 1 FROM FRAME_RES_MODEL T WHERE T.MODEL_CODE = 'ITSM.基础数据' AND
T.MODEL_ID = A1.Model_Id);

--清除模块信息
SELECT * FROM FRAME_RES_MODEL T WHERE T.MODEL_CODE = 'ITSM.基础数据';
DELETE FROM FRAME_RES_MODEL T WHERE T.MODEL_CODE = 'ITSM.基础数据';

```

G.单据编码规则

将以下脚本查找出来的数据导出成insert脚本即可.

```

-- 替换成实际编号,不导出CURRENTVALUE(当前编码值)
select FORMCODE_RULEID,
       FORMNAME,
       FORMCODE,
       PREFIX,
       CYCLE,
       SERIALNUM,
       FACTORYCODE,
       DELETEFLAG,
       ENABLEFLAG,
       REMARK,
       VERSION,
       COMPANYCODE,
       ISPREFIX,
       USECYCLE,
       VERSION_,
       LOCKED_STATUS_,
       DELETE_STATUS_,
       CREATE_DATE_,
       CREATE_USER_,
       UPDATE_DATE_,
       UPDATE_USER_,
       TENANT_CODE_,
       DATA_SECRET_LEVCODE_,
       DATA_TENANT_ID_,
       DATA_ORG_ID_,
       DATA_ORG_CODE_,
       DATA_ORG_LEVCODE_,
       DATA_DEPT_ID_,
       DATA_DEPT_CODE_,
       DATA_DEPT_LEVCODE_
from T_V01_FORMCODE_RULE
where FORMCODE = 'BGDH';

```

```
-- 替换成实际编号,导出CURRENTVALUE
select *
from T_V01_FORMCODE_RULE
where FORMCODE = 'BGDH';
-- 清理
delete from T_V01_FORMCODE_RULE where FORMCODE in ('OR_CORR_FORM','OR');
```

H.APP信息

导出

将以下脚本查找出来的数据导出成insert脚本即可.

```
-- 替换成实际APP_CODE
select *
from frame_app_info
where APP_CODE_ = 'OperationReport';
-- APP_ORG
select *
from FRAME_APP_ORG
where APP_ID_ in (select OID
                  from FRAME_APP_INFO
                  where APP_CODE_ = 'OperationReport');
-- 功能列表
select t.*
from frame_app_function t
      join frame_app_info t1 on t1.oid = t.app_id_
where t1.oid in (select OID
                 from frame_app_info
                 where APP_CODE_ = 'OperationReport');
-- 菜单
select t.*
from frame_app_menu_list t
      join frame_app_info t1 on t1.oid = t.appid
where t1.oid in (select OID
                 from frame_app_info
                 where APP_CODE_ = 'OperationReport');
-- 常量
select *
from FRAME_APP_DATA_CONSTANT
where APP_CODE_ in (select FRAME_APP_INFO.APP_CODE_
                   from frame_app_info
                   where APP_CODE_ = 'OperationReport');
```

清理

```
-- 替换成实际APP_CODE
delete
from frame_app_info
where APP_CODE_ = 'OperationReport';
```

```

-- APP_ORG
delete
from FRAME_APP_ORG
where APP_ID_ in (select OID
                  from FRAME_APP_INFO
                  where APP_CODE_ = 'OperationReport');

-- 功能列表
delete frame_app_function
where OID in (
    select t.OID
    from frame_app_function t
        join frame_app_info t1 on t1.oid = t.app_id_
    where t1.oid in (select OID
                    from frame_app_info
                    where APP_CODE_ = 'OperationReport'));

-- 菜单
delete frame_app_menu_list
where MENULIST_ID in (
    select t.MENULIST_ID
    from frame_app_menu_list t
        join frame_app_info t1 on t1.oid = t.appid
    where t1.oid in (select OID
                    from frame_app_info
                    where APP_CODE_ = 'OperationReport'));

```

J.数据隔离策略

清理脚本

```

-- 清理数据隔离
-- 清理数据隔离策略
delete from FRAME_DATA_PRIV_TYPE where PRIVTYPE_CODE='89';
delete from FRAME_DATA_PRIVILEGE_DEF
where PRIVILEGETYPE_ID in (
    select PRIVTYPE_ID
    from FRAME_DATA_PRIV_TYPE
    where PRIVTYPE_CODE = '89'
);

-- 清理数据表
delete from FRAME_DB_TBLGROUP where TBLGROUP_NAME='运维报告';
delete from FRAME_DB_TABLE_INFO
where TBLGROUP_ID in (
    select FRAME_DB_TBLGROUP.TBLGROUP_ID
    from FRAME_DB_TBLGROUP
    where TBLGROUP_NAME = '运维报告');

-- 导出数据表隔离策略
delete from FRAME_DB_TBL_DEFAULT_PRIV
where TABLE_ID in (
    select FRAME_DB_TABLE_INFO.TABLE_ID

```



```

from FRAME_DB_TABLE_INFO
where TBLGROUP_ID in (
    select FRAME_DB_TBLGROUP.TBLGROUP_ID
    from FRAME_DB_TBLGROUP
    where TBLGROUP_NAME = '运维报告')
);
delete from FRAME_DB_TBLPRIVILEGE_REL
where TABLE_ID in (
    select FRAME_DB_TABLE_INFO.TABLE_ID
    from FRAME_DB_TABLE_INFO
    where TBLGROUP_ID in (
        select FRAME_DB_TBLGROUP.TBLGROUP_ID
        from FRAME_DB_TBLGROUP
        where TBLGROUP_NAME = '运维报告')
);
-- 授权相关
delete from FRAME_ROLE_INFO where CODE_ in ('ROLE_YJYWRY', 'ROLE_YWBG-GLY', 'ROLE_YWBG-
JDZHZRR', 'ROLE_YWBG-YDZHZRR', 'ROLE_YWBGLD', 'ROLE_YWBG_PTRY', 'opsreport');

delete from FRAME_AUTH_ROLE_FUNCSREVICE
where ROLE_ID in (
    select OID
    from FRAME_ROLE_INFO
    where CODE_ in
        ('ROLE_YJYWRY', 'ROLE_YWBG-GLY', 'ROLE_YWBG-JDZHZRR', 'ROLE_YWBG-YDZHZRR',
'ROLE_YWBGLD', 'ROLE_YWBG_PTRY',
        'opsreport')
);

delete from FRAME_AUTH_ROLE_PAGE where ROLE_ID in (
    select OID
    from FRAME_ROLE_INFO
    where CODE_ in
        ('ROLE_YJYWRY', 'ROLE_YWBG-GLY', 'ROLE_YWBG-JDZHZRR', 'ROLE_YWBG-YDZHZRR',
'ROLE_YWBGLD', 'ROLE_YWBG_PTRY',
        'opsreport')
);

delete from FRAME_AUTH_ROLE_PAGEFUNC where ROLE_ID in (
    select OID
    from FRAME_ROLE_INFO
    where CODE_ in
        ('ROLE_YJYWRY', 'ROLE_YWBG-GLY', 'ROLE_YWBG-JDZHZRR', 'ROLE_YWBG-YDZHZRR',
'ROLE_YWBGLD', 'ROLE_YWBG_PTRY',
        'opsreport')
);

delete from FRAME_AUTH_SERVICE_TBL
where TABLE_ID in (
    select FRAME_DB_TABLE_INFO.TABLE_ID
    from FRAME_DB_TABLE_INFO
    where TBLGROUP_ID in (
        select FRAME_DB_TBLGROUP.TBLGROUP_ID

```

```

        from FRAME_DB_TBLGROUP
        where TBLGROUP_NAME = '运维报告')
    )
    or AUTH_FUNCSERVICE_ID in (
        select FRAME_AUTH_ROLE_FUNCSREVICE.AUTH_FUNCSERVICE_ID
        from FRAME_AUTH_ROLE_FUNCSREVICE
        where ROLE_ID in (
            select OID
            from FRAME_ROLE_INFO
            where CODE_ in
                ('ROLE_YJYWRY', 'ROLE_YWBG-GLY', 'ROLE_YWBG-JDZHRR', 'ROLE_YWBG-
YDZHRR', 'ROLE_YWBGLD',
                'ROLE_YWBG_PTRY', 'opsreport')
        )
    );

delete from FRAME_AUTH_TBL_PRIVILEGES
where AUTH_SERVICETBL_ID in (
    select FRAME_AUTH_SERVICE_TBL.AUTH_SERVICETBL_ID
    from FRAME_AUTH_SERVICE_TBL
    where TABLE_ID in (
        select FRAME_DB_TABLE_INFO.TABLE_ID
        from FRAME_DB_TABLE_INFO
        where TBLGROUP_ID in (
            select FRAME_DB_TBLGROUP.TBLGROUP_ID
            from FRAME_DB_TBLGROUP
            where TBLGROUP_NAME = '运维报告')
        )
    or AUTH_FUNCSERVICE_ID in (
        select FRAME_AUTH_ROLE_FUNCSREVICE.AUTH_FUNCSERVICE_ID
        from FRAME_AUTH_ROLE_FUNCSREVICE
        where ROLE_ID in (
            select OID
            from FRAME_ROLE_INFO
            where CODE_ in
                ('ROLE_YJYWRY', 'ROLE_YWBG-GLY', 'ROLE_YWBG-JDZHRR', 'ROLE_YWBG-
YDZHRR', 'ROLE_YWBGLD',
                'ROLE_YWBG_PTRY', 'opsreport')
        )
    )
);

```

导出脚本

```

-- 导出数据隔离
select * from FRAME_DATA_PRIV_TYPE where PRIVTYPE_CODE='89';
select *
from FRAME_DATA_PRIVILEGE_DEF
where PRIVILEGETYPE_ID in (
    select PRIVTYPE_ID
    from FRAME_DATA_PRIV_TYPE
    where PRIVTYPE_CODE = '89'
);

```

```

);
-- 导出数据表
select * from FRAME_DB_TBLGROUP where TBLGROUP_NAME='运维报告';
select * from FRAME_DB_TABLE_INFO
where TBLGROUP_ID in (
    select FRAME_DB_TBLGROUP.TBLGROUP_ID
    from FRAME_DB_TBLGROUP
    where TBLGROUP_NAME = '运维报告');
-- 导出数据表隔离策略
select * from FRAME_DB_TBL_DEFAULT_PRIV
where TABLE_ID in (
    select FRAME_DB_TABLE_INFO.TABLE_ID
    from FRAME_DB_TABLE_INFO
    where TBLGROUP_ID in (
        select FRAME_DB_TBLGROUP.TBLGROUP_ID
        from FRAME_DB_TBLGROUP
        where TBLGROUP_NAME = '运维报告')
);
select * from FRAME_DB_TBLPRIVILEGE_REL
where TABLE_ID in (
    select FRAME_DB_TABLE_INFO.TABLE_ID
    from FRAME_DB_TABLE_INFO
    where TBLGROUP_ID in (
        select FRAME_DB_TBLGROUP.TBLGROUP_ID
        from FRAME_DB_TBLGROUP
        where TBLGROUP_NAME = '运维报告')
);
-- 授权相关
select * from FRAME_ROLE_INFO where CODE_ in ('ROLE_YWBG_PTRY', 'ROLE_YWBG-
YDZHRR', 'ROLE_YWBG-JDPGBGZRR', 'ROLE_YWBG-JDYWBGZRR', 'ROLE_YWBG-JDZHRR', 'ROLE_YWBG-
GLY', 'ROLE_YWBGLD', 'ROLE_YWBG-
LSZHRR', 'ROLE_YXXHSPBM', 'ROLE_YWBG-WTZGFQR', 'opsreport', 'ROLE_YWBG-NDZHRR');

select * from FRAME_AUTH_ROLE_FUNCSREVICE
where ROLE_ID in (
    select OID
    from FRAME_ROLE_INFO
    where CODE_ in
        ('ROLE_YWBG_PTRY', 'ROLE_YWBG-YDZHRR', 'ROLE_YWBG-JDPGBGZRR', 'ROLE_YWBG-
JDYWBGZRR', 'ROLE_YWBG-JDZHRR', 'ROLE_YWBG-GLY', 'ROLE_YWBGLD', 'ROLE_YWBG-
LSZHRR', 'ROLE_YXXHSPBM', 'ROLE_YWBG-WTZGFQR', 'opsreport', 'ROLE_YWBG-NDZHRR')
);

select * from FRAME_AUTH_ROLE_PAGE where ROLE_ID in (
    select OID
    from FRAME_ROLE_INFO
    where CODE_ in
        ('ROLE_YWBG_PTRY', 'ROLE_YWBG-YDZHRR', 'ROLE_YWBG-JDPGBGZRR', 'ROLE_YWBG-
JDYWBGZRR', 'ROLE_YWBG-JDZHRR', 'ROLE_YWBG-GLY', 'ROLE_YWBGLD', 'ROLE_YWBG-
LSZHRR', 'ROLE_YXXHSPBM', 'ROLE_YWBG-WTZGFQR', 'opsreport', 'ROLE_YWBG-NDZHRR')
);

select * from FRAME_AUTH_ROLE_PAGEFUNC where ROLE_ID in (

```

```

select OID
from FRAME_ROLE_INFO
where CODE_ in
    ('ROLE_YWBG_PTRY', 'ROLE_YWBG-YDZHRR', 'ROLE_YWBG-JDPGBGZRR', 'ROLE_YWBG-
JDYWBGZRR', 'ROLE_YWBG-JDZHRR', 'ROLE_YWBG-GLY', 'ROLE_YWBGLD', 'ROLE_YWBG-
LSZHRR', 'ROLE_YXXHSPBM', 'ROLE_YWBG-WTZGFQR', 'opsreport', 'ROLE_YWBG-NDZHRR')
);

select * from FRAME_AUTH_SERVICE_TBL
where TABLE_ID in (
    select FRAME_DB_TABLE_INFO.TABLE_ID
    from FRAME_DB_TABLE_INFO
    where TBLGROUP_ID in (
        select FRAME_DB_TBLGROUP.TBLGROUP_ID
        from FRAME_DB_TBLGROUP
        where TBLGROUP_NAME = '运维报告')
)
or AUTH_FUNCSERVICE_ID in (
    select FRAME_AUTH_ROLE_FUNCSEVICE.AUTH_FUNCSERVICE_ID
    from FRAME_AUTH_ROLE_FUNCSEVICE
    where ROLE_ID in (
        select OID
        from FRAME_ROLE_INFO
        where CODE_ in
            ('ROLE_YWBG_PTRY', 'ROLE_YWBG-YDZHRR', 'ROLE_YWBG-JDPGBGZRR', 'ROLE_YWBG-
JDYWBGZRR', 'ROLE_YWBG-JDZHRR', 'ROLE_YWBG-GLY', 'ROLE_YWBGLD', 'ROLE_YWBG-
LSZHRR', 'ROLE_YXXHSPBM', 'ROLE_YWBG-WTZGFQR', 'opsreport', 'ROLE_YWBG-NDZHRR')
    )
);

select * from FRAME_AUTH_TBL_PRIVILEGES
where AUTH_SERVICETBL_ID in (
    select FRAME_AUTH_SERVICE_TBL.AUTH_SERVICETBL_ID
    from FRAME_AUTH_SERVICE_TBL
    where TABLE_ID in (
        select FRAME_DB_TABLE_INFO.TABLE_ID
        from FRAME_DB_TABLE_INFO
        where TBLGROUP_ID in (
            select FRAME_DB_TBLGROUP.TBLGROUP_ID
            from FRAME_DB_TBLGROUP
            where TBLGROUP_NAME = '运维报告')
    )
or AUTH_FUNCSERVICE_ID in (
    select FRAME_AUTH_ROLE_FUNCSEVICE.AUTH_FUNCSERVICE_ID
    from FRAME_AUTH_ROLE_FUNCSEVICE
    where ROLE_ID in (
        select OID
        from FRAME_ROLE_INFO
        where CODE_ in
            ('ROLE_YWBG_PTRY', 'ROLE_YWBG-YDZHRR', 'ROLE_YWBG-
JDPGBGZRR', 'ROLE_YWBG-JDYWBGZRR', 'ROLE_YWBG-JDZHRR', 'ROLE_YWBG-
GLY', 'ROLE_YWBGLD', 'ROLE_YWBG-
LSZHRR', 'ROLE_YXXHSPBM', 'ROLE_YWBG-WTZGFQR', 'opsreport', 'ROLE_YWBG-NDZHRR')
    )
);

```

```
)  
)  
);
```

K.业务升级脚本
