## Ovation 3.2 之 44 号补丁的分析

Zhu Wei Aug, 2014 Rev01

## 一,引言

在 CAD+Solaris 的 Ovation 时代,CB 编译出的逻辑关系都是存在于 Oracle 数据库的 CPI/BPI 关系表中;在 CAD+Windows 的 Ovation 时代,CB 编译出的逻辑关系是生成了相应的 ocs 文件;而到了 svg 时代,CB 编译出的逻辑关系则是生成了相应的 xml 文件。

## 二,内容

Ovation 3.2 的 44 号补丁更新了 CB 宏的编译结果。一个 svg 的宏在通过编译后会 生成 mcr 文件与 tft 文件,而 mcr 文件是标明逻辑关系的文件。如下图,左侧是 安装了 OV3200044 补丁后编译出的 mcr 文件,格式稍微有些不同。

```
<q cb-classname="ControlAlgorithm" id="4087"</pre>
                                                         <q cb-classname="ControlAlgorithm" id="4087"</pre>
      cb-function-name="AND"
                                                                cb-function-name="AND"
      cb-modify-id="9"
                                                                cb-modify-id="9"
                                                                cb-alg-in1="$720$0UT$"
cb-alg-in2="$720$0UT$"
      cb-alg-in1="$720$cb-alg-out$
      cb-alg-in2="$720$cb-alg-out$"
                                         installed
      cb-alq-out="D40874"
                                                                cb-alg-out="D40871"
<g cb-classname="ControlAlgorithm" id="758"</pre>
                                                         <q cb-classname="ControlAlgorithm" id="758"</pre>
      cb-function-name="NOT"
                                                                cb-function-name="NOT"
      cb-modify-id="1"
                                                                cb-modify-id="1"
      cb-alg-in1="$4087$cb-alg-out$"
                                                                cb-alg-in1="$4087$0UT$"
<g cb-classname="ControlAlgorithm" id="225"</pre>
                                                         <g cb-classname="ControlAlgorithm" id="225"</pre>
      cb-function-name="INPUT"
                                                                cb-function-name="INPUT"
      cb-modify-id="1"
                                                                cb-modify-id="1"
      cb-alg-out="TEST"
                                                                cb-alg-out="TEST"
```

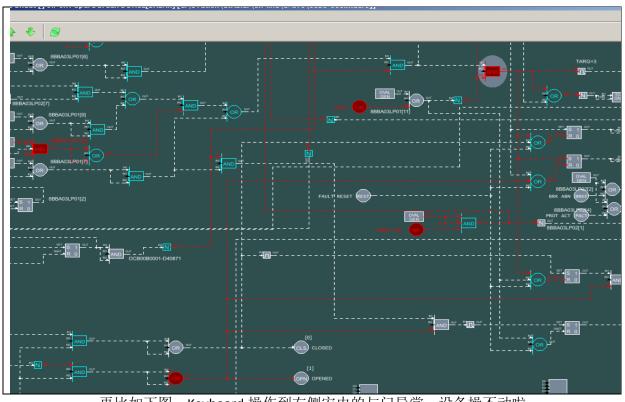
ControlFunctions 目录下的 svg 逻辑图,编译后都会生成相应的 xml 与 htm 文件,而 Load 控制器时,是将 xml 文件 Load 下去。由于 44 号补丁需要安装在所有的 Ovation 上位机,这样就会有 4 种情况,生成最终的使用宏的那张控制逻辑 (control sheet)的 xml。

- (1) 没有安装 44 补丁的站编译了宏,且编译了使用宏的逻辑图;
- (2) 安装有44补丁的站编译了宏,且编译了使用宏的逻辑图:
- (3) 没有安装 44 补丁的站编译了宏,安装 44 补丁的站编译了使用宏的逻辑图 (这里也可以表示,安装 44 补丁后没有重新编译宏,但是重新编译了使用宏的逻辑图);
- (4) 安装 44 补丁的站编译了宏,没有安装 44 补丁的站编译了使用宏的逻辑图。

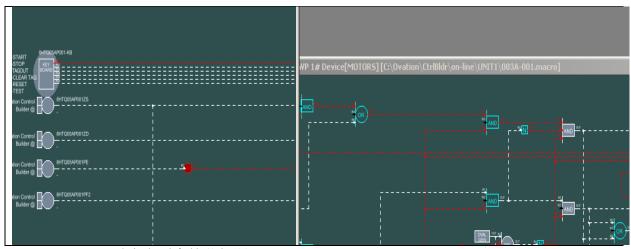
比较这 4 种情况生成的 xml 文件,会发现 1/2/4 生成的 xml 文件是完全一样的,除了时间戳。而 3 生成的 xml 文件就不太一样了,在宏模板展开的时候会丢点,如下图,左侧就是第 3 种情况,编译出的 xml 文件居然丢点了。

```
// COULT OTLL OAL QUITUREL ACCTOUS
                                                                                           </ControlProgramInstruction>
        <ControlProgramInstruction function-name="ONESHOT" function-id="-1" algor</p>
                                                                                           <ControlProgramInstruction function-name="ONESHOT" function-id="-1" algorithm-</pre>
"001-25" version-id="2" >
                                                                                    001-25" version-id="2" >
                <ControlProgramArgument type="0" point="0CB003A801-C36581" />
                                                                                                   <ControlProgramArqument type="0" point="0CB003A001-C36581</p>
                <ControlProgramArgument type="6" point="" /> point missed!
                                                                                                   <ControlProgramArgument type='0" point="8HTQ05AP001LP01" bit="0" />
                <ControlProgramArgument type='0" point="" />
                                                                                                   <ControlProgramArgument type=""" point="" />
                <ControlProgramArgument type="0" point=""/>
                                                                                                   <ControlProgramArgument type="0" point="" />
                <ControlProgramArgument type="0" point="0CB003A001-D36582" />
                                                                                                   <ControlProgramArqument type="0" point="OCB003A001-D36582" />
        </ControlProgramInstruction>
                                                                                           </ControlProgramInstruction>
        <ControlProgramInstruction function-name="PACK16" function-id="-1" algori</p>
                                                                                           <ControlProgramInstruction function-name="PACK16" function-id="-1" algorithm-i</pre>
001-28" version-id="2" >
```

这样就会出现各种意外情况,比如下面的,左下角的或门,右上角的与门都出问 题啦。



再比如下图,Keyboard 操作到右侧宏内的与门异常,设备操不动啦。



那么如何避免这些问题呢?

所有上位机都要安装 44 号补丁,补丁安装完成后,先把宏编译一下,再将所有用 到宏的逻辑图编译一下(使用 Force Reload 强制编译,当然编译前不要忘了 Reconcile),然后再下装控制器(下装前控制器最好再做一次 Reconcile,然后 Clear&Load,应该先下装主控制器)。另外既然正确情况下,生成的 xml 除了时 间戳外完全一致,那么在编译完成后,我们完全可以去验证逻辑图的正确性。使

用文件比较软件或者自行写一个文件比较的程序都是可以的(这样就确保万无一失啦)。