

You must allocate your CNTL data set with the appropriate name:

<hlq>.V92D8nnn.CNTL, where nnn is the Julian date of the day you are running the SDW.

You can either

1. send SAS Technical Support the setup.zip file from the following directory in your depot:

<depot home directory>/products/sasinstall\_\_92330\_\_mvs\_\_ne\_\_sp0\_\_1

(the numbers in red may be different, look for “sasinstall” and “mvs” in the directory). Be sure to transfer the file in binary mode! A technical support consultant will edit the file and return it to you.

2. Or edit the file yourself. Inside that zip file is a file named SASFoundation/9.2/sasinstall\_install.xml. Make the changes shown below in red (this is just a snippet of the big file – search on “allocate-cntl” to find it). The basic idea is to commenting out all of the actions inside the “allocate-cntl” target, but we have to leave the target in there or other sections will not run. A comment in XML starts with <!-- and ends with -->, but you can’t have nested comments so we have to end comments and restart them when there are comments already in the XML.

```
<!-- Allocate the control data set -->
```

```
<target name="allocate-cntl" depends="install-action">
  <echo message="Allocating
${zos.high.level.qualifier}.V92D8${day.qualifier}.CNTL data set"/>
  <!--
  <exec executable="/bin/tso"
    outputproperty="delete.out"
    resultproperty="delete.result">
    <arg value="-t"/>
    <arg value="del
'${zos.high.level.qualifier}.V92D8${day.qualifier}.CNTL'"/>
  </exec>
  -->
  <!-- create SMS parameters for ALLOC only if present -->
```

```

<!--
    <condition property="parm.storclas"
value="storclas(${zos.sms.sc})" else="">
        <and>
            <not>
                <and>
                    <isset property="zos.sms"/>
                    <isfalse value="${zos.sms}"/>
                </and>
            </not>
            <isset property="zos.sms.sc"/>
            <not>
                <length string="${zos.sms.sc}" trim="true" length="0"/>
            </not>
        </and>
    </condition>

    <condition property="parm.mgmtclas"
value="mgmtclas(${zos.sms.mc})" else="">
        <and>
            <not>
                <and>
                    <isset property="zos.sms"/>
                    <isfalse value="${zos.sms}"/>
                </and>
            </not>
            <isset property="zos.sms.mc"/>
            <not>
                <length string="${zos.sms.mc}" trim="true" length="0"/>
            </not>
        </and>
    </condition>

```

```

        <condition property="parm.dataclas"
value="dataclas(${zos.sms.dc})" else="">
    <and>
        <not>
            <and>
                <isset property="zos.sms"/>
                <isfalse value="${zos.sms}"/>
            </and>
        </not>
        <isset property="zos.sms.dc"/>
        <not>
            <length string="${zos.sms.dc}" trim="true" length="0"/>
        </not>
    </and>
</condition>

```

```

        <condition property="parm.voldisk"
value="vol(${zos.job.voldisk})" else="">
    <and>
        <isset property="zos.job.voldisk"/>
        <not>
            <length string="${zos.job.voldisk}" trim="true"
length="0"/>
        </not>
    </and>
</condition>

```

```

<condition property="parm.sms-alloc" value="" else="%SMS.">
    <or>
        <and>
            <isset property="zos.sms"/>

```

```

        <isfalse value="${zos.sms}"/>
    </and>
    <and>
        <not>
            <isset property="zos.sms"/>
        </not>
        <isset property="zos.job.voldisk"/>
    </and>
</or>
</condition>
-->
<!-- allocate control data set -->
<!--
<exec executable="/bin/tso"
    outputproperty="allocate-cntl.out"
    resultproperty="allocate-cntl.result">
    <arg value="-t"/>
    <arg value="listc
ent('${zos.high.level.qualifier}.V92D8${day.qualifier}.CNTL') all"/>
    <env key="TSOALLOC" value="SYSUT1"/>
    <env key="SYSUT1" value="alloc
da('${zos.high.level.qualifier}.V92D8${day.qualifier}.CNTL) new catalog
dsorg(po)
        recfm(f,b) lrecl(80) blksize(6160) space(1000,250)
dir(150)
        ${parm.storclas} ${parm.mgmtclas} ${parm.dataclas}
${parm.voldisk} dsntype(pds)"/>
</exec>
<fail message="***allocate-cntl target Failed">
    <condition>
        <not>
            <equals arg1="${allocate-cntl.result}" arg2="0" />
        </not>

```

```
        </condition>
    ${allocate-cntl.out}
        </fail>
-->
    </target>
```