

Update instructions FS 9.11.8 -> 9.11.16 received from Ed Himwich

A. The old (DDC) rack types dbbc and dbbc/fila10g are now dbbc_ddc and dbbc_ddc/fila10g, respectively. For PFB, there are new rack types dbbc_pfb and dbbc_pfb/fila10g for PFB without and with FiLa10G, respectively.

B. The line that use to be the number of CoMos (just after PFB version line and just before DBBC IF conversion factors), should now have up four fields that represent the number of Cores on a given CoMo, in in order A-D. The number of fields imply the number of CoMo, for example three fields implies you only have IFs A, B, and C. Acceptable field values are 0-4. The sum of the field values must be four or less. For example for one Core on each of four CoMos. you would use:

```
1 1 1 1
```

for three CoMos with 2 Cores on IF C, you would use:

```
1 1 2
```

C. You may want to change the FiLa10G input select (last line) to vsi1-2 for support of 32 channel modes.

7. **(optional) "fix_define"**: Have you ever had trouble with procedure contents being logged repeatedly in the same log file? They should not be. This is due in at least in some cases to the "define" lines in .prc files being damaged (possibly from manual editing, instead of with PFMED). There is a new utility:

```
/usr2/fs/misc/fix_define
```

that can fix this. You can run it when the FS is **not active**:

```
cd /usr2/proc  
/usr2/fs/misc/fix_define *.prc
```

and it will fix the "define" lines in a minimalist way. It will let pass anything that will work unless you use the "-f", which will force everything into canonical form.

1. If you have a local "sterp" or "erchk" program, the method for passing the error information to these programs has changed. If your local version uses the previous get_err() routine to retrieve the error information, you will probably be able to just copy the new version of this routine from

```
/usr2/fs/st.default/st-0.0.0/sterp/get_err.c
```

or

```
/usr2/fs/st.default/erchk/get_err.c
```

(they are identical) into your source directory and recompile. If this doesn't work for you, the changes needed should be fairly simple and I am happy to help.

2. Station procedures for the DBBC have now been unified for DDC/PFB and continuous/non-continuous cal and are independent of the channels recorded for the mode. To fully utilize this, changes to your local procedures are needed. The specific standard procedures that have changed are:

```
ifman
ifagc
preob
caltsys
iread
bread
```

Examples of the new versions of these procedures can be found in

```
/usr2/fs/st.default/proc/dstation.prc
```

(Hopefully it all works. I can imagine there might be some rough edges in real world use.)

Additionally, if you have other personality/cal dependencies in your station procedures. You can use the new "if" command (thanks to Jon for suggesting this) to handle the dependencies. Please see the help page for the "if" command ("help=if" in oprin) for more information. Also please free to contact me.

This is the first implementation of these features. Please send your feedback on how they can be improved.

3. Fila10G configuration initialization is now integrated into FMSET. This is intended to make this easier to manage and to reduce the chances of forgetting to send the initialization before sync-ing. This is supported with the '/usr2/control/fila10g_cfg.ctl' file. An example file can be found in

```
/usr2/fs/st.default/control/fila10g_cfg.ctl
```

which basically implements the (uncustomized) contents of the old example "fila10g_cfg" procedure. You should set up your own version with the configurations you want, customized for your station. The syntax of the file is described more fully in the example file, but the basic features are:

- up to 21 different configurations can be defined
- each configuration definition starts with a "\$config name" line
- "name" is a unique string of up to 16 non-space characters to identify the configuration
- after the "\$config name" line there can be an arbitrary number of commands for the set-up of that configuration
- the commands are "raw" fila10g commands, i.e., no leading "fila10g=", maximum command length is 128 characters
- embedded white-space is allowed in the commands, but trailing white space is trimmed
- all lines starting with asterisks ("*") are comments, which can appear anywhere
- no blank lines or lines that start with white space are allowed
- an empty file and configurations with no commands are allowed (but not of any clear value)

In FMSET, if you select 's' and after confirming that you do want to sync and whether you want to also sync the DBBC, you will be offered a list of the available named configurations. You can select the configuration by entered its displayed number, or enter "0" or just hit <Enter> for none.

If a configuration was selected (and after the optional, but recommended, DBBC synchronization), the name will be logged and then each command will be individually logged and executed. The idea of logging the commands is to provide a record and to make debugging easier if there is an error. Execution of the commands will stop if there is an error. Finally, if all the commands are executed without errors, the fila10g will be sync'd as usual.

This is the first implementation of this feature. Please send your feedback on how this can be improved.

4. A new keyword is available in skedf.ctl: "vsi_align" to control whether (and how) for DBBC_PFB and DDC_PFB/FILA10G systems, DRUDG inserts an dbbc=vsi_align=... command, after "form=flex' in each setup procedure. The keyword supports values:

- none - do not insert a dbbc=vsi_align=... command
- 0 - insert dbbc=vsi_align=0 command
- 1 - insert dbbc=vsi_align=1 command
- ask - prompt for each schedule whether to use "none", "1", or "2"

If the keyword is no present, the effect is the same as "none".

An example can be found in:

`/usr2/fs/st.default/control/skedf.ctl`

I am not sure what the correct setting is for this option, but it may be "1", at least in some cases. There may need to be more discussion about this.