NSLS-II CSX Beamline Docs Documentation

Release 0.1

Stuart B. Wilkins

November 29, 2015

Contents:

CHAPTER

ONE

CSX-1 (23-ID-1) BEAMLINE DOCCUMENTATION

Contents:

1.1 Fast CCD Detector

1.1.1 Introduction

The FastCCD installed in the endstation at CSX-1 is of the LBNL Fast CCD design. The sensor contains 1920 x 960 pixels of 30 µm x 30 µm and is arranged into two halves of 960 rows by 960 columns with the columns parallel to the long CCD axis. There is one output for each 10 columns (a "super column") which results in 192 individual outputs and analogue to digital converters (ADC). The CCD camera can either be used in a traditional CCD with an x-ray shutter exposing the full chip, or in a framestore (frame transfer) mode by covering two quarters of the CCD with a light (x-ray) block effectively exposing half the chip along the column direction.

The analogue CCD signal is digitized by a custom designed fCRIC. Each fCRIC has 16 analogue inputs and digitizes with 13 bit precision and had 16 bit dynamic range. This is accomplished by having 3 gain ranges of 8x, 4x and 1x with an auto gain feature. In order to allow negative charge injection. The ADC is biased at a value of approximately 4096 (0x1000 in hex) with the exact value dependent on the ADC channel. The gain settings are stored in the two most significant bits of each ADC reading. The schematic of a single fCRIC chanel is shown in the *LBNL fCRIC Circuit Diagram*.

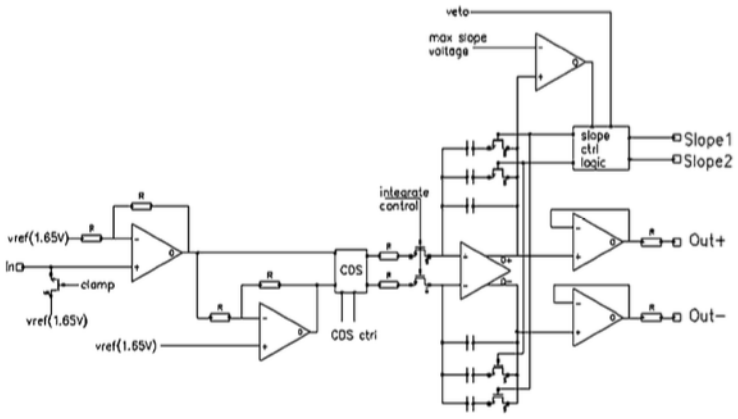


Fig. 1.1: LBNL fCRIC Circuit Diagram

The specifications of the CCD are summarized below:

- Pixel Size: 30 µm x 30 µm
- Active Area: 1920 pixels (column) x 960 pixels (row)
- 192 super columns = 192 outputs (480 rows x 10 columns)
- Back illuminated
- 250 µm - 350 µm thickness
- Full well : ~900k e^- per pixel
- Sensitivity : 6 e^- / ADU for 8x gain (max gain)
- Pixel readout time: 500 µs
- Digitization time: 2 µs at 120 Hz
- 100 Hz maximum data collection

1.1.2 Data Format

In treating the raw CCD data from the FastCCD there are a few important considerations related to the multi-gain behaviour of the fCRIC amplifier and digitizer. The raw 16 bit values that are recorded in the data file follow the *16 Bit fCRIC Data Format* shown below with the two gain bits following the *fCRIC Gain Setting*.

Table 1.1: 16 Bit fCRIC Data Format

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
G1	G0	ERR	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00

Table 1.2: fCRIC Gain Setting

G1	G0	Gain	Pre-factor
0	0	x8	x1
1	0	x2	x4
1	1	x1	x8

Here the two most significant bits record the gain setting for the encoded value. The least significant 13 bits hold the measured analogue value. The actual value is therefore related to the measured value by:

$$A_{\mathrm{corr}} = G(A_{\mathrm{meas}} - O)$$

where A_{corr} is the corrected intensity, A_{meas} is the measured value by the ADC, G is the gain of the ADC and O is the bias offset.

1.1.3 Dark Image Subtraction

Due to the multi gain nature of the fCRIC it is therefore necessary to take 3 dark images at different gain settings to obtain the different ADC offsets under these modes. As the lower gain settings are not subject to considerable contribution due to dark current it is usually justifiable to measure only the highest gain dark image repeatedly. Given 3 dark images for the different gain settings the images the following python pseudo code can be used to correct for dark current and gain:

```
import numpy as np

def subtract_background(image, dark_image, gain = [1, 4, 8]):
    gain_mask_8 = (image & 0xC000) == 0xC000
    gain_mask_4 = (image & 0xC000) == 0x8000
    gain_mask_1 = (image & 0xC000) == 0x0000
```

```
    cor_image = image.astype(np.float16)
    cor_image -= gain_mask_8 * dark_image[2]
    cor_image -= gain_mask_4 * dark_image[1]
    cor_image -= gain_mask_1 * dark_image[0]

    gain_image = (gain_mask_8 * gain[2]) + (gain_mask_4 * gain[1]) + (gain_mask_1 * gain[0])

    return (cor_image * gain_image), gain_image
```

1.1.4 Useful Links

- LBNL Fast CCD Site
- csxtools python analysis routines
- libcin low level c driver
- areaDetector Driver

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

THREE

HELP!! THE %$^$#@% JUST CRASHED

3.1 Managing IOCs

Soft IOCs are managed through the `manage-iocs` script. To obtain a list of softiocs running on a NSLS-II computer use the command `manage-iocs report` an example is shown below for `xf23id1-ioc3`:

```
[swilkins@xf23id1-ioc3 ~]$ manage-iocs report
nBASE            | IOC             | USER            |  PORT | EXEC
/epics/iocs      | apcupsd         | root            |  5000 | /epics/iocs/apcupsd/st.cmd
/epics/iocs      | cam-diag1       | softioc         |  4202 | /epics/iocs/cam-diag1/st.cmd
/epics/iocs      | cam-diag6       | softioc         |  4300 | /epics/iocs/cam-diag6/st.cmd
/epics/iocs      | cam-dif1        | softioc         |  4204 | /epics/iocs/cam-dif1/st.cmd
/epics/iocs      | cam-dif2        | softioc         |  4205 | /epics/iocs/cam-dif2/st.cmd
/epics/iocs      | cam-dif3        | softioc         |  4206 | /epics/iocs/cam-dif3/st.cmd
/epics/iocs      | cam-dif-beam    | softioc         |  4201 | /epics/iocs/cam-dif-beam/st.cmd
/epics/iocs      | ct-eps          | softioc         |  4002 | /epics/iocs/ct-eps/st.cmd
/epics/iocs      | es-dg645        | softioc         |  5013 | /epics/iocs/es-dg645/st.cmd
/epics/iocs      | es-K2611        | softioc         |  4302 | /epics/iocs/es-K2611/st.cmd
/epics/iocs      | es-tctrl1       | softioc         |  5010 | /epics/iocs/es-tctrl1/st.cmd
/epics/iocs      | es-vortex       | softioc         |  4301 | /epics/iocs/es-vortex/st.cmd
/epics/iocs      | mc11            | softioc         |  5001 | /epics/iocs/mc11/st.cmd
/epics/iocs      | mc12            | softioc         |  5002 | /epics/iocs/mc12/st.cmd
/epics/iocs      | mc13            | softioc         |  5003 | /epics/iocs/mc13/st.cmd
/epics/iocs      | omegaM4061      | softioc         |  5012 | /epics/iocs/omegaM4061/st.cmd
/epics/iocs      | simdetector     | softioc         |  4203 | /epics/iocs/simdetector/st.cmd
/epics/iocs      | simmotor        | softioc         |  8001 | /epics/iocs/simmotor/st.cmd
/epics/iocs      | timestamp       | softioc         |  6001 | /epics/iocs/timestamp/st.cmd
/epics/iocs      | va-bakeout-01   | softioc         |  4001 | /epics/iocs/va-bakeout-01/st.cmd
/epics/iocs      | zebra           | softioc         |  5011 | /epics/iocs/zebra/st.cmd
```

To connect to the IOC console, telnet to localhost at the port that is shown in the table. For example to connect to the `mc12` console issue the command:

```
[swilkins@xf23id1-ioc3 ~]$ telnet localhost 5002
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
@@@ Welcome to procServ (procServ Process Server 2.6.0)
@@@ Use ^X to kill the child, auto restart is ON, use ^T to toggle auto restart
@@@ procServ server PID: 10584
@@@ Server startup directory: /epics/iocs/mc12
@@@ Child startup directory: /epics/iocs/mc12
@@@ Child "mc12" started as: /epics/iocs/mc12/st.cmd
@@@ Child "mc12" PID: 28044
```

```
@@@ procServ server started at: Tue Oct 20 17:35:25 2015
@@@ Child "mc12" started at: Fri Nov 13 12:49:49 2015
@@@ 0 user(s) and 0 logger(s) connected (plus you)
```

In order to reboot the IOC, type `[CTRL] + X`. To leave the console type `[CTRL] + ]` and type `close` at the `telnet>` prompt

To start all IOCs configured on the system issue the command `sudo manage-iocs startall` and if needed to stop all IOCs issue the command `sudo manage-iocs stopall`

3.2 OLog Glassfish Server

To reboot the glassfish server on `xf23id-ca.cs.nsls2.local` execute:

```
swilkins@xf23id-ca:~$sudo su - glassfish
glassfish@xf23id-ca:~$cd glassfish3/bin/
glassfish@xf23id-ca:~/glassfish3/bin$ ./asadmin stop-domain domain1
glassfish@xf23id-ca:~/glassfish3/bin$ ./asadmin stop-domain domain2
glassfish@xf23id-ca:~/glassfish3/bin$ ./asadmin start-domain domain1
glassfish@xf23id-ca:~/glassfish3/bin$ ./asadmin start-domain domain2
```

CHAPTER

FOUR

CONTROLS ACCOUNT SETUP GUIDE

4.1 Introduction

NSLS-II controls accounts allow access to all computers at CSX

CHAPTER

FIVE

INSTALLING A PERSONAL CONDA ENVIRONMENT

5.1 Introduction

This guide covers installing the conda packaging system *miniconda* into your own space. This has a number of advantages, including full controll on package versions. Due to the large nature of the files, it is suggested not to install into your home directory if you are using NFS home directories. At CSX conda should be installed on the GPFS file system in the directory `/GPFS/xf23id/users/<uid>` where `<uid>` is your user ID.

To create a directory on the **GPFS** system, do the following:

```
[swilkins@23id1-srv2 ~]$ sudo mkdir /GPFS/xf23id/users/<uid>
[swilkins@23id1-srv2 ~]$ sudo chown <uid> /GPFS/xf23id/users/<uid>
[swilkins@23id1-srv2 ~]$ sudo chmod 750 /GPFS/xf23id/users/<uid>
```

Where `<uid>` is your userid.

5.2 Installing Miniconda

Install the latest *miniconda*. This can be done by downloading the latest miniconda binary installer from conda.pydata.org.

Once the file is downloaded make the file executable and run the installer. **DO NOT** let the installer change your `.bashrc` file.:

```
[swilkins@xf23id1-srv2 ~/Downloads]$ chmod u+x Miniconda3-latest-Linux-x86_64.sh
[swilkins@xf23id1-srv2 ~/Downloads]$ ./Miniconda3-latest-Linux-x86_64.sh
```

When prompted, enter the GPFS path for the install location, for example:

```
Do you approve the license terms? [yes|no]
[no] >>> yes

Miniconda3 will now be installed into this location:
/home/swilkins/miniconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/swilkins/miniconda3] >>> /GPFS/xf23id/users/swilkins/miniconda3
PREFIX=/GPFS/xf23id/users/swilkins/miniconda3
```

To add the *miniconda* to yor path, edit your `.bashrc` file with your favorite editor and add the following lines.

```
if [ -e "/GPFS/xf23id/users/swilkins/miniconda3" ]; then
    export PATH="/GPFS/xf23id/users/swilkins/miniconda3/bin:$PATH"
fi
```

Substituting `swilkins` for your user ID. To enable the path, now source your `.bashrc` file:

```
[swilkins@xf23id1-srv2 ~/Downloads]$ source ~/.bashrc
```

5.3 Installing a Custom Conda Environment

Once miniconda is installed and in the path, configure conda to use the NSLS-II anaconda cloud server:

```
[swilkins@xf23id1-srv2 ~/Downloads]$ conda install anaconda-client conda-build --yes
[swilkins@xf23id1-srv2 ~/Downloads]$ conda config --add channels anaconda
[swilkins@xf23id1-srv2 ~/Downloads]$ conda config --add channels latest
[swilkins@xf23id1-srv2 ~/Downloads]$ conda config --add create_default_packages pip
[swilkins@xf23id1-srv2 ~/Downloads]$ conda config --add create_default_packages anaconda-client
[swilkins@xf23id1-srv2 ~/Downloads]$ anaconda config --set url https://conda.nsls2.bnl.gov/api
[swilkins@xf23id1-srv2 ~/Downloads]$ conda config --remove channels defaults --force
[swilkins@xf23id1-srv2 ~/Downloads]$ conda update --all --yes
```

Congratulations! You now have a personal installation of *miniconda* connected to the NSLS-II anaconda cloud server. Now you can create a new environment for doing your analysis. To create and activate the environment type:

```
[swilkins@xf23id1-srv2 ~/Downloads]$ conda create -n analysis python=3.5
[swilkins@xf23id1-srv2 ~/Downloads]$ source activate analysis
(analysis)[swilkins@xf23id1-srv2 ~/Downloads]$ conda install dataportal
(analysis)[swilkins@xf23id1-srv2 ~/Downloads]$ conda install ipython-notebook
```

5.4 Installing a Custom Notebook Kernel

To get your custom environment to work with the notebook server, you have to create a kernal file in your `.python/kernel` directory (where *my-analysis-kernel* can be any name you wish to know this kernel by):

```
(analysis)[swilkins@xf23id1-srv2 ~]$ cd ~/.ipython/kernels/
(analysis)[swilkins@xf23id1-srv2 ~/.ipython/kernels]$ mkdir my-analysis-kernel
```

Now create a *kernel json file* to let the notebook server know how to run this kernel. Create a file `kernel.json` in the directory `my-analysis-kernel` with your favorite text editor such as:

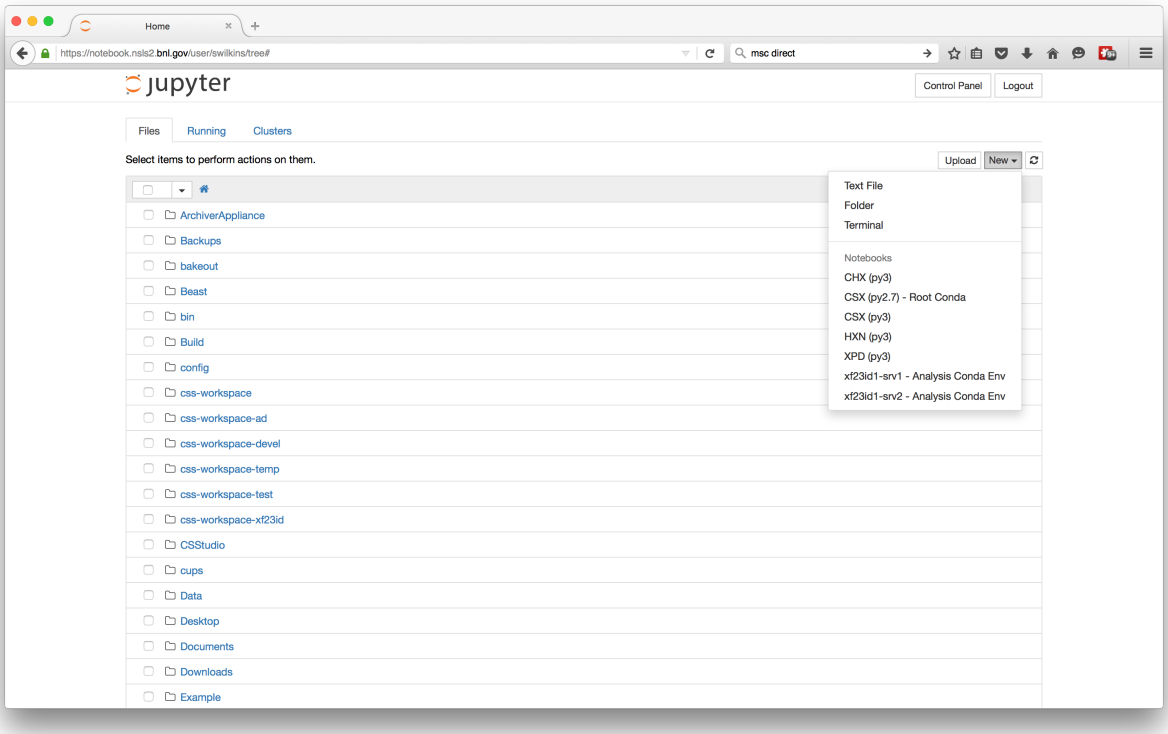
```
{"argv":["/home/swilkins/miniconda3/envs/analysis/bin/python3.5",
         "-m", "IPython.kernel", "-f", "{connection_file}"],
  "display_name":"xf23id1-srv2 - Analysis Conda Env",
  "host":"xf23id1-srv2"
}
```

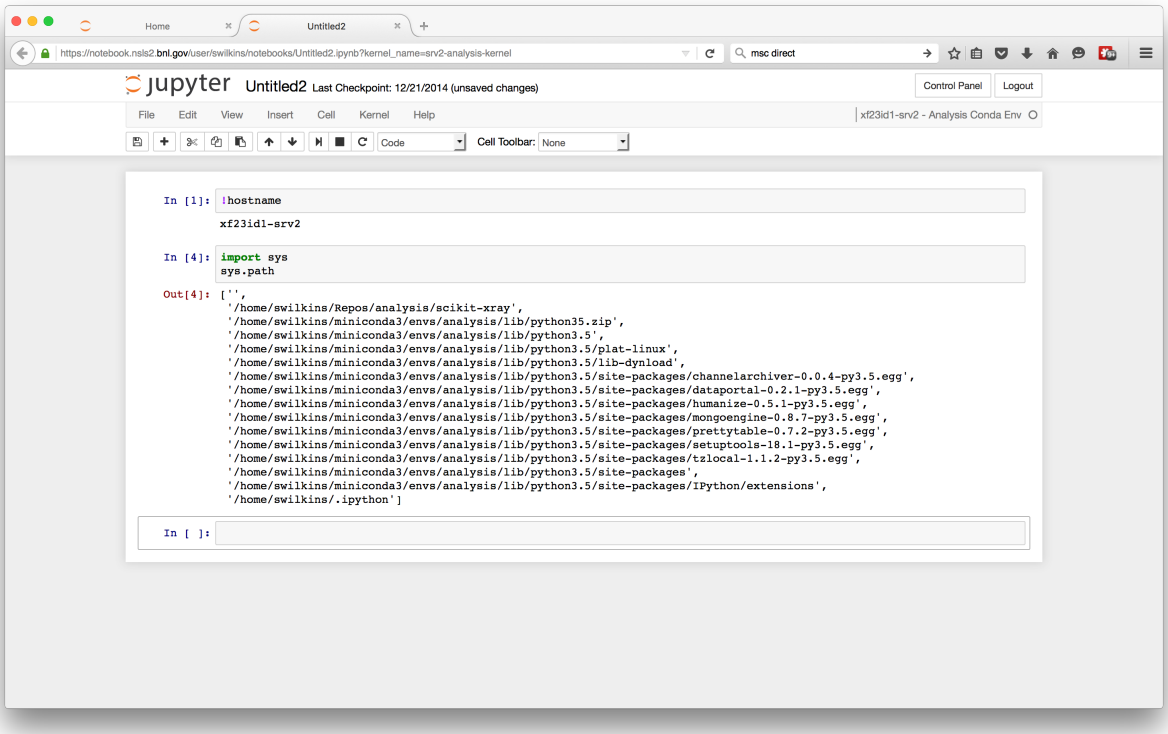
Where the path `/home/swilkins/miniconda3/envs/analysis/bin/python3.5` should point to the path of python in your home directory conda environment. `display_name` should be a nice name for this kernel, and the `host` is the computer on which the kernel should run.

If all works OK, the new kernel shouls show up in the kernel list on notebook.nsls2.bnl.gov

Running a new notebook from that option will now run a kernel in the new conda environment:

Congratulations!!


Home
https://notebook.nsls2.bnl.gov/user/swilkins/tree#
msc direct
jupyter
Control Panel
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
Text File
Folder
Terminal
Notebooks
CHX (py3)
CSX (py2.7) - Root Conda
CSX (py3)
HXN (py3)
XPD (py3)
xf23id1-srv1 - Analysis Conda Env
xf23id1-srv2 - Analysis Conda Env
ArcherAppliance
Backups
bakeout
Beast
bin
Build
config
css-workspace
css-workspace-ad
css-workspace-devel
css-workspace-temp
css-workspace-test
css-workspace-xf23id
CSStudio
cups
Data
Desktop
Documents
Downloads
Example


Home
Untitled2
https://notebook.nsls2.bnl.gov/user/swilkins/notebooks/Untitled2.ipynb?kernel_name=srv2-analysis-kernel
msc direct
jupyter
Untitled2
Last Checkpoint: 12/21/2014 (unsaved changes)
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Help
xf23id1-srv2 - Analysis Conda Env
Code
Cell Toolbar:
None
In [1]: !hostname
xf23id1-srv2
In [4]: import sys
sys.path
Out[4]: ['',
'/home/swilkins/Repos/analysis/scikit-xray',
'/home/swilkins/miniconda3/envs/analysis/lib/python35.zip',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/plat-linux',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/lib-dynload',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages/channelarchiver-0.0.4-py3.5.egg',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages/dataportal-0.2.1-py3.5.egg',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages/humanize-0.5.1-py3.5.egg',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages/mongoengine-0.8.7-py3.5.egg',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages/prettytable-0.7.2-py3.5.egg',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages/setuptools-18.1-py3.5.egg',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages/tzlocal-1.1.2-py3.5.egg',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages',
'/home/swilkins/miniconda3/envs/analysis/lib/python3.5/site-packages/IPython/extensions',
'/home/swilkins/.ipython']
In [ ]:

CHAPTER

SIX

VIM COMMANDS CHEAT SHEET

6.1 How to Exit

:q[uit]	Quit Vim. This fails when changes have been made.
:q[uit]!	Quit without writing.
:cq[uit]	Quit always, without writing.
:wq	Write the current file and exit.
:wq!	Write the current file and exit always.
:wq {file}	Write to {file}. Exit if not editing the last
:wq! {file}	Write to {file} and exit always.
:[range]wq[!]	[file] Same as above, but only write the lines in [range].
ZZ	Write current file, if modified, and exit.
ZQ	Quit current file and exit (same as ":q!").

6.2 Editing a File

:e[dit]	Edit the current file. This is useful to re-edit the current file, when it has been changed outside of Vim.
:e[dit]!	Edit the current file always. Discard any changes to the current buffer. This is useful if you want to start all over again.
:e[dit] {file}	Edit {file}.
:e[dit]! {file}	Edit {file} always. Discard any changes to the current buffer.
gf	Edit the file whose name is under or after the cursor. Mnemonic: "goto file".

6.3 Inserting Text

a	Append text after the cursor [count] times.
A	Append text at the end of the line [count] times.
i	Insert text before the cursor [count] times.
I	Insert text before the first non-blank in the line [count] times.
gI	Insert text in column 1 [count] times.
o	Begin a new line below the cursor and insert text, repeat [count] times.
O	Begin a new line above the cursor and insert text, repeat [count] times.

6.4 Inserting a file

:r[ead] [name]	Insert the file [name] below the cursor.
:r[ead] !{cmd}	Execute {cmd} and insert its standard output below the cursor.

6.5 Deleting Text

<Del> or x	Delete [count] characters under and after the cursor
X	Delete [count] characters before the cursor
d{motion}	Delete text that {motion} moves over
dd	Delete [count] lines
D	Delete the characters under the cursor until the end of the line
{Visual}x or {Visual}d	Delete the highlighted text (for {Visual} see *Selecting Text*).
{Visual}CTRL-H or {Visual}	When in Select mode: Delete the highlighted text
{Visual}X or {Visual}D	Delete the highlighted lines
:[range]d[elete]	Delete [range] lines (default: current line)
:[range]d[elete] {count}	Delete {count} lines, starting with [range]

6.6 Changing (or Replacing) Text

r{char}	replace the character under the cursor with {char}.
R	Enter Insert mode, replacing characters rather than inserting
~	Switch case of the character under the cursor and move the cursor to the right. If a [count] is given, do that many characters.
~{motion}	switch case of {motion} text.
{Visual}~	Switch case of highlighted text

6.7 Substituting

:[range]s[ubstitute]/{pattern}/{string}/[c][e][g][r][i][I] [count]	For each line in [range] replace a match of {pattern} with {string}.
:[range]s[ubstitute] [c][e][g][r][i][I] [count] :[range]&[c][e][g][r][i][I] [count]	Repeat last :substitute with same search pattern and substitute string, but without the same flags. You may add extra flags

```
The arguments that you can use for the substitute commands:
[c]  Confirm each substitution.  Vim positions the cursor on the matching
  string.  You can type:
      'y'      to substitute this match
      'n'      to skip this match
         to skip this match
      'a'      to substitute this and all remaining matches {not in Vi}
      'q'      to quit substituting {not in Vi}
      CTRL-E  to scroll the screen up {not in Vi}
      CTRL-Y  to scroll the screen down {not in Vi}.
[e]     When the search pattern fails, do not issue an error message and, in
  particular, continue in maps as if no error occurred.
[g]  Replace all occurrences in the line.  Without this argument,
  replacement occurs only for the first occurrence in each line.
[i]  Ignore case for the pattern.
[I]  Don't ignore case for the pattern.
[p]  Print the line containing the last substitute.
```

6.8 Copying and Moving Text

"{a-zA-Z0-9.%#:-"}	Use register {a-zA-Z0-9.%#:-"} for next delete, yank or put (use uppercase character to append with delete and yank) ({.%#:} only work with put).
:reg[isters]	Display the contents of all numbered and named registers.
:reg[isters] {arg}	Display the contents of the numbered and named registers that are mentioned in {arg}.
:di[splay] [arg]	Same as :registers.
["x]y{motion}	Yank {motion} text [into register x].
["x]yy	Yank [count] lines [into register x]
["x]Y	yank [count] lines [into register x] (synonym for yy).
{Visual}["x]y	Yank the highlighted text [into register x] (for {Visual} see *Selecting Text*).
{Visual}["x]Y	Yank the highlighted lines [into register x]
:[range]y[ank] [x]	Yank [range] lines [into register x].
:[range]y[ank] [x] {count}	Yank {count} lines, starting with last line number in [range] (default: current line), [into register x].
["x]p	Put the text [from register x] after the cursor [count] times.
["x]P	Put the text [from register x] before the cursor [count] times.
["x]gp	Just like "p", but leave the cursor just after the new text.
["x]gP	Just like "P", but leave the cursor just after the new text.
:[line]pu[t] [x]	Put the text [from register x] after [line] (default current line).
:[line]pu[t]! [x]	Put the text [from register x] before [line] (default current line).

6.9 Undo/Redo/Repeat

u	Undo [count] changes.
:u[ndo]	Undo one change.
CTRL-R	Redo [count] changes which were undone.
:red[o]	Redo one change which was undone.
U	Undo all latest changes on one line. {Vi: while not moved off of it}
.	Repeat last change, with count replaced with [count].

6.10 Moving Around

```
Basic motion commands:

        k
     h   l
        j
```

h or

[count] characters to the left (exclusive).

l or or

[count] characters to the right (exclusive).

k or or CTRL-P

[count] lines upward

j or or CTRL-J or or CTRL-N

[count] lines downward (linewise).

0

To the first character of the line (exclusive).

<Home>

To the first character of the line (exclusive).

^

To the first non-blank character of the line

$ or <End>

To the end of the line and [count - 1] lines downward

g0 or g<Home>

When lines wrap ('wrap on): To the first character of the screen line (exclusive). Differs from "0" when a line is wider than the screen. When lines don't wrap ('wrap' off): To the leftmost character of the current line that is on the screen. Differs from "0" when the first character of the line is not on the screen.

g^

When lines wrap ('wrap' on): To the first non-blank character of the screen line (exclusive). Differs from "^" when a line is wider than the screen. When lines don't wrap ('wrap' off): To the leftmost non-blank character of the current line that is on the screen. Differs from "^" when the first non-blank character of the line is not on the screen.

g$ or g<End&gr;

When lines wrap ('wrap' on): To the last character of the screen line and [count - 1] screen lines downward (inclusive). Differs from "$" when a line is wider than the screen. When lines don't wrap ('wrap' off): To the rightmost character of the current line that is visible on the screen. Differs from "$" when the last character of the line is not on the screen or when a count is used.

f{char}

To [count]'th occurrence of {char} to the right. The cursor is placed on {char} (inclusive).

F{char}

To the [count]'th occurrence of {char} to the left. The cursor is placed on {char} (inclusive).

t{char}

Till before [count]'th occurrence of {char} to the right. The cursor is placed on the character left of {char} (inclusive).

T{char}

Till after [count]'th occurrence of {char} to the left. The cursor is placed on the character right of {char} (inclusive).

;

Repeat latest f, t, F or T [count] times.

,

Repeat latest f, t, F or T in opposite direction [count] times.

- <minus>

[count] lines upward, on the first non-blank character (linewise).

- or

  CTRL-M or <CR>

[count] lines downward, on the first non-blank character (linewise).

_ <underscore>

[count] - 1 lines downward, on the first non-blank character (linewise).

<C-End> or G

Goto line [count], default last line, on the first non-blank character.

<C-Home> or gg

Goto line [count], default first line, on the first non-blank character.

<S-Right> or w

[count] words forward

<C-Right> or W

[count] WORDS forward

e

Forward to the end of word [count]

E

Forward to the end of WORD [count]

<S-Left> or b

[count] words backward

<C-Left> or B

[count] WORDS backward

ge

Backward to the end of word [count]

gE

Backward to the end of WORD [count]

These commands move over words or WORDS.

A word consists of a sequence of letters, digits and underscores, or a sequence of other non-blank characters, separated with white space (spaces, tabs,). This can be changed with the 'iskeyword' option.

A WORD consists of a sequence of non-blank characters, separated with white space. An empty line is also considered to be a word and a WORD.

(	[count] sentences backward
)	[count] sentences forward
{	[count] paragraphs backward
}	[count] paragraphs forward
]]	[count] sections forward or to the next '{' in the first column. When used after an operator, then the '}' in the first column.
][	[count] sections forward or to the next '}' in the first column
[[	[count] sections backward or to the previous '{' in the first column
[]	[count] sections backward or to the previous '}' in the first column

Screen movement commands

26.	Center the screen on the cursor
zt	Scroll the screen so the cursor is at the top
zb	Scroll the screen so the cursor is at the bottom

6.11 Marks

m{a-zA-Z}	Set mark {a-zA-Z} at cursor position (does not move the cursor, this is not a motion command).
m' or m'	Set the previous context mark. This can be jumped to with the "'" or "`" command (does not move the cursor, this is not a motion command).
:[range]ma[rk] {a-zA-Z}	Set mark {a-zA-Z} at last line number in [range], column 0. Default is cursor line.
:[range]k{a-zA-Z}	Same as :mark, but the space before the mark name can be omitted.
'{a-z}	To the first non-blank character on the line with mark {a-z} (linewise).
'{A-Z0-9}	To the first non-blank character on the line with mark {A-Z0-9} in the correct file
`{a-z}	To the mark {a-z}
`{A-Z0-9}	To the mark {A-Z0-9} in the correct file
:marks	List all the current marks (not a motion command).
:marks {arg}	List the marks that are mentioned in {arg} (not a motion command). For example:

6.12 Searching

/{pattern}[/]	Search forward for the [count]'th occurrence of {pattern}
/{pattern}/{offset}	Search forward for the [count]'th occurrence of {pattern} and go {offset} lines up or down.
/<CR>	Search forward for the [count]'th latest used pattern
//{offset}<CR>	Search forward for the [count]'th latest used pattern with new. If {offset} is empty no offset is used.
?{pattern}[?]<CR>	Search backward for the [count]'th previous occurrence of {pattern}
?{pattern}?{offset}<CR>	Search backward for the [count]'th previous occurrence of {pattern} and go {offset} lines up or down
?<CR>	Search backward for the [count]'th latest used pattern
??{offset}<CR>	Search backward for the [count]'th latest used pattern with new {offset}. If {offset} is empty no offset is used.
n	Repeat the latest "/" or "?" [count] times.
N	Repeat the latest "/" or "?" [count] times in opposite direction.

6.13 Selecting Text (Visual Mode)

To select text, enter visual mode with one of the commands below, and use *motion commands* to highlight the text you are interested in. Then, use some command on the text.

```
The operators that can be used are:
  ~  switch case
  d  delete
  c  change
  y  yank
  >  shift right
  <  shift left
  !  filter through external command
  =  filter through 'equalprg' option command
  gq  format lines to 'textwidth' length
```

v	start Visual mode per character.
V	start Visual mode linewise.
<Esc>	exit Visual mode without making any changes

6.14 How to Suspend

CTRL-Z	Suspend Vim, like ":stop". Works in Normal and in Visual mode. In Insert and Command-line mode, the CTRL-Z is inserted as a normal character.
:sus[pend][!] or :st[op][!]	Suspend Vim. If the '!' is not given and 'autowrite' is set, every buffer with changes and a file name is written out. If the '!' is given or 'autowrite' is not set, changed buffers are not written, don't forget to bring Vim back to the foreground later!

|image0|

Daniel Gryniewicz / dang@fprintf.net

CHAPTER

SEVEN

DOWNLOADS

Download the CSX Documentation as a PDF

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search