

iSCP

What is iSCP

The Integrated SLAC Control Program (iSCP) is a collection of interactive accelerator physics application programs. Using iSCP on a Unix workstation you can run the VMS SCP, any EPICS display, plus a number of other programs such as EPICS extensions.

iSCP is based on the Eclipse framework. At the time of writing iSCP is mostly just a collection of scripts that can be used with the "External programs" facility of Eclipse. Extensive help on this facility of Eclipse is available in Eclipse's Help system (See help on "External Programs"), and in books on Eclipse. See [References](#).

In the future we may take advantage of the facility of Eclipse to act as a framework, or skeleton, for interactive applications programs themselves, making use of Eclipse's Rich Client Platform (RCP). That is not covered here.

iSCP Requirements

iSCP is implemented in Unix. The launch scripts have been tested on RHEL 9 Linux running from AFS, and that is the configuration described below. Able users should also be able to edit the startup script (iscp) and the launch scripts (*.eosi) to run in Solaris or Mac OS X.

Prerequisites To Use iSCP

To use iSCP smoothly you should first have passwordless authentication to the hosts you wish to use. Nominally this should be at least:

Host	As User	Use
lcls-dev	laci	LCLS EPICS displays
PEP-II "Production" (gtw00)	cddev	PEP-II EPICS displays, and most interactive programs like alarm watchdog
AFS PEP-II "Prod on Dev" (tersk etc)	yourself	xterms
VMS	yourself	The SCP

Please see <http://www.slac.stanford.edu/grp/cd/soft/unix/dev/slaonly/rsasetup.html> for instructions for passwordless authentication to those systems.

(Check Jingchen's doc helps you with authentication for yourself and for cddev).

(Check Jingchen's page helps with unix and VMS).

Starting iSCP

We have a basic iSCP, that includes the LCLS and PEP-II displays, plus a number of the common programs in the control system already configured. That setup is in `/afs/slac.stanford.edu/g/cd/soft/support/eclipse_ws/basic`.

Later, as iSCP is used in a production setting, we will create a single production instance of Eclipse for iSCP and provide help for running it.

One-time setup, create an Eclipse workspace

Log into a Redhat Linux machine, and check you have an AFS token. If you do not already have an area for your eclipse workspaces, create one (for instance `~/eclipse_ws`). And then **copy the basic workspace to that area**. Please, do not use the directory `/afs/slac.stanford.edu/g/cd/soft/support/eclipse_ws/basic` as your own eclipse workspace, since your instance of Eclipse will write there. instead, be sure to copy it over to you own space. For example:

```
u/cd/greg> mkdir -p ~/eclipse_ws
u/cd/greg> cd eclipse_ws
u/cd/greg> cp -R /afs/slac.stanford.edu/g/cd/soft/support/eclipse_ws/basic basic
```

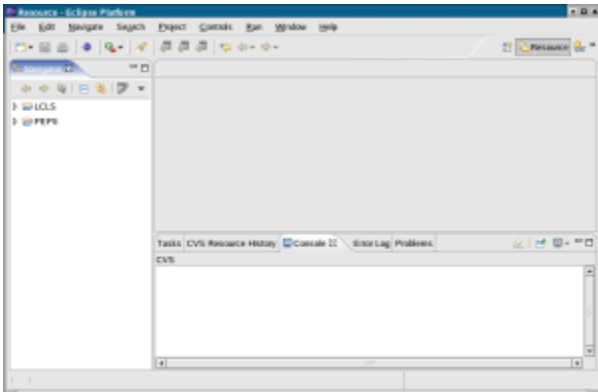
Start iSCP

Note, since we do not yet have a release area for Linux, the iscp script is just run from its reference area:

```
u/cd/greg> /afs/slac/g/cd/soft/ref/app/iscp/script/iscp
```

When Eclipse starts, it will ask you to enter a workspace directory. Enter the full pathname of the directory you are using as your workspace. In the example above it would be `/afs/slac.stanford.edu/u/cd/greg/eclipse_ws/basic`. *Note: the default might be `/afs/slac.stanford.edu/u/cd/greg/eclipse_ws/workspace`, so you should change workspace to basic if following the example exactly.*

Once the eclipse starts, with this workspace, it will display a screen like:



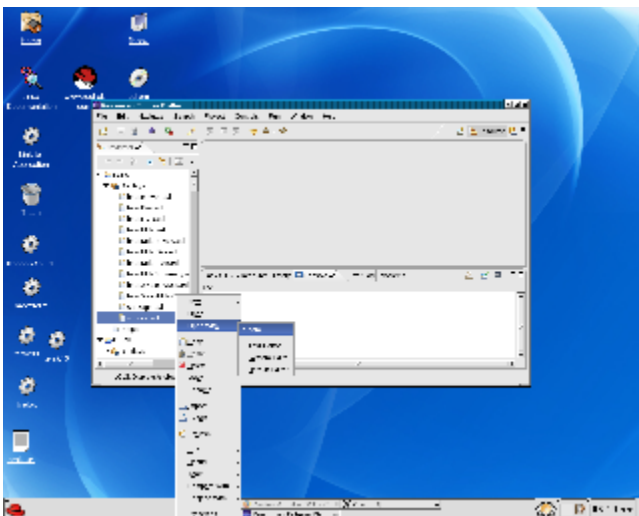
You can now proceed to launch applications and displays, and if you like, to configure Eclipse to launch other applications.

Launching Displays

To launch a display, first expand one of the accelerator projects, PEPII or LCLS, in the Navigator View (the pane on the left-hand-side is called the "Navigator" view). You may, if you have been wondering around Eclipse, have found your way out of the Resource Perspective so that there is no Navigator View on the left. If that has happened, select the Window menu -> Open Perspective -> Navigator (or Other ... Navigator).

Having expanded, for instance LCLS, you should see a subfolder "Displays", and if you expand that, a list of Displays.

To launch an EDM display, select it, then Right-click -> Open With -> edm.



PEPII displays (dm and dm2k) have different launchers, depending on the host on which they should launch. In those cases, for instance for a dm2k display, you may have to choose between `dm2k_pack`, or `dm2k_pepii` to launch the display.

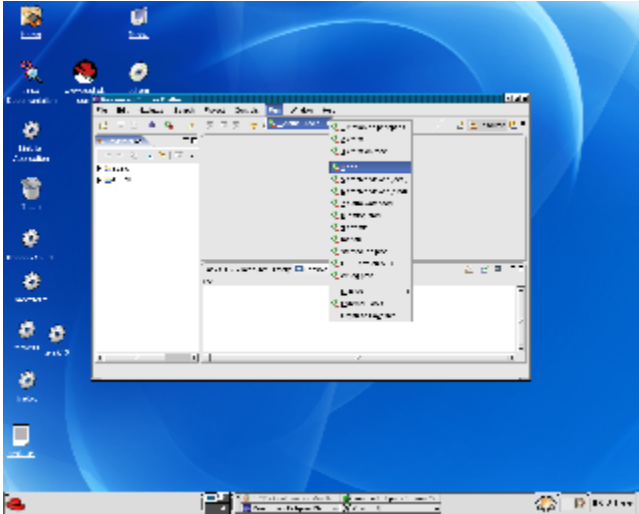
Working Set "Accelerator Operations"

To cut down on the long list of PEPII displays offered, you can select only those that might be interesting to Accelerator Operations by selecting the triangle in the top right corner of the Navigator view, and selecting "Accelerator Operations". You can add or remove displays from the list of those in that working set, using "Edit Active Working Set".

Launching Programs.

To launch a program, like the SCP, you must have passwordless authentication to the machine on which it will be launched. Check the table above, or the External Programs setup (see below) directly, if you're not sure what host that will be.

For instance, to run the SCP on MCC (for which you must have passwordless rsa2 authentication):



This section describes how you configure your Eclipse session to launch external control system programs. The basic workspace (described above) already has in it the configurations for LCLS and PEP-II displays, and some like the VMS SCP, striptool, etc. This section describes how that was achieved as a reference.

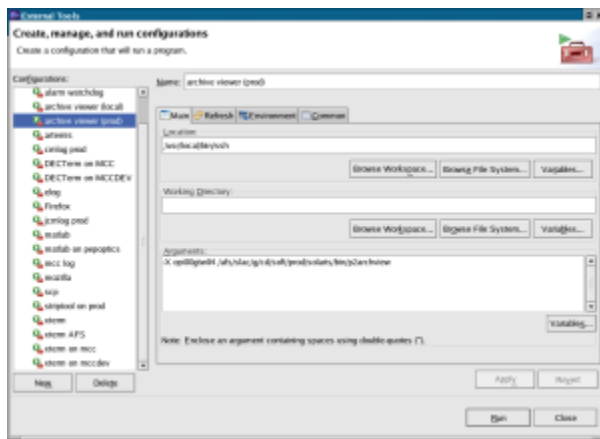
Operating System Independent Launching in iSCP

The launch configuration scripts are in CVS under

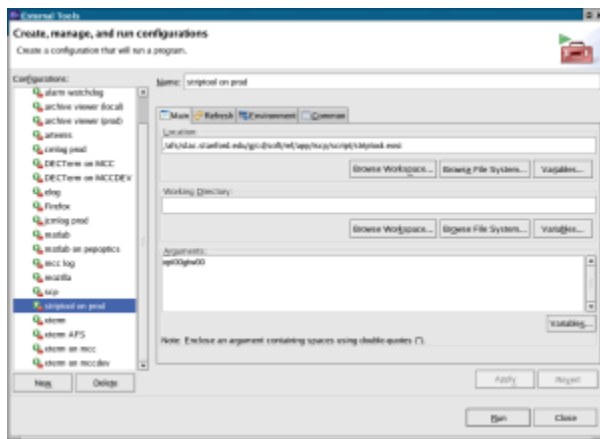
So, configuring your Eclipse session to launch control system programs is just a matter of teaching it to run those eosi scripts. That configuration is done through the Run menu, which is available from the Resource Perspective.

If you are not in it, first switch to the Resource perspective (Window->Open Perspective and select "Resource"). Then select Run->External Tools->External Tools Select Program in the left hand window, and click "New". Then fill in the dialog as show for each tool you wish to configure.

This configuration logs into MCC, starts a DECTerm, and in that Decterm starts a SCP. Note that there are some significant bugs in running a SCP from RHEL 9. [see the list below](#).

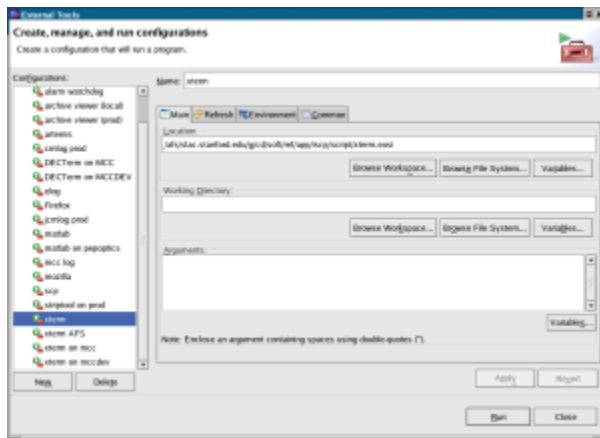


Striptool



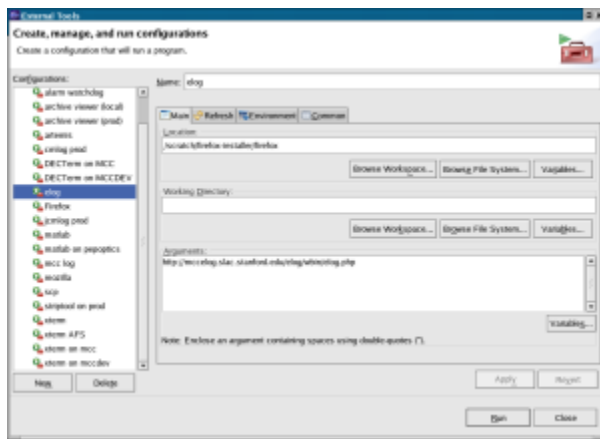
xterm on localhost

This configuration just launches an xterm on the host on which you started Eclipse. That is, this only works if you're on some variant of Unix.



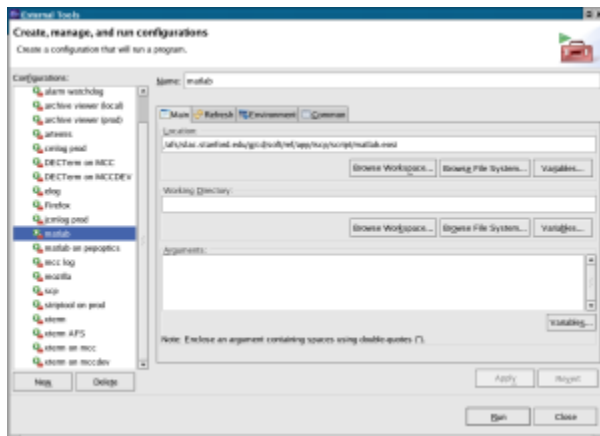
xterm on AFS

This configuration starts an xterm and logs you into a tersk machine.



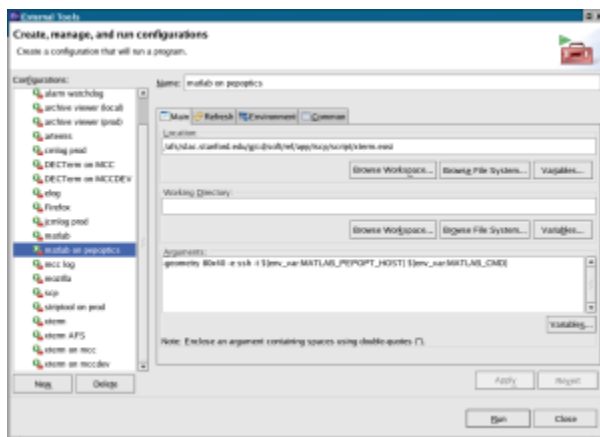
Matlab

This configuration starts matlab on the host on which you have started iscp. The matlab session is started right in Eclipse. If you prefer to use matlab from an xterm command prompt, or even use the matlab GUI, create your own version of the matlab.eosi file for the behavior you want.



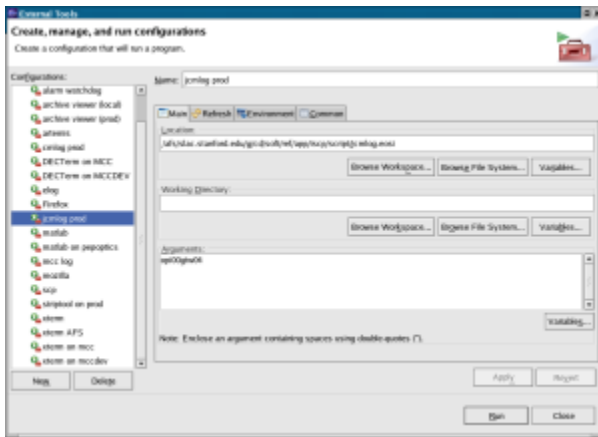
Matlab on pepoptics

This configuration starts an xterm, logs you into pepoptics, and starts matlab there. This should be used very sparingly, since pepoptics should only be used for immediate beam optimization problems.



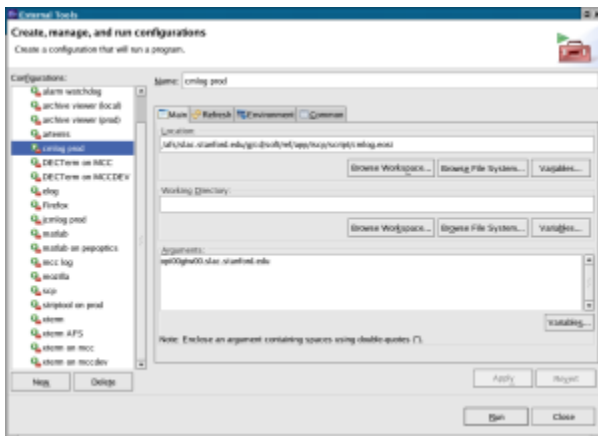
jcmlog

This configuration starts the new jcmlog message browser on gtw04, which is the gateway host able to receive cmlog messages for prod and "dev on prod" pepii applications. Note that the host is passed as an argument to the eos file.



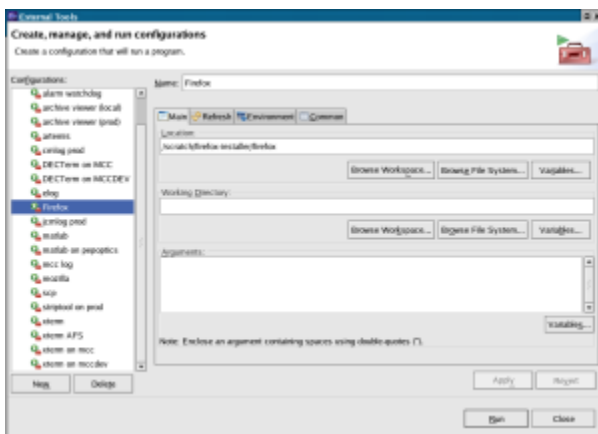
cmlog

This configuration starts the cmlog message browser on gtw00.



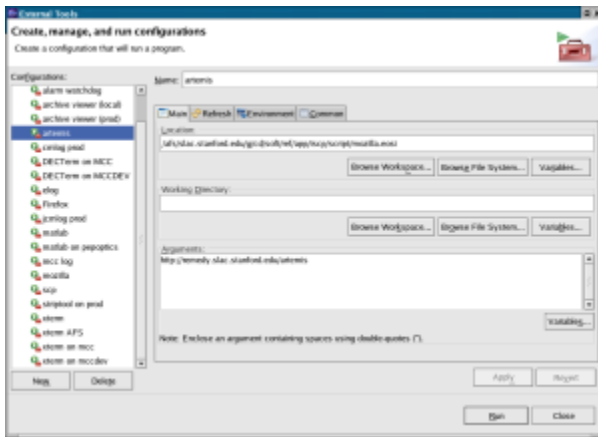
Firefox

This configuration just starts firefox (if it's available) on your host, as an alternative method of starting this browser.



Artemis

This configuration starts mozilla on your local host (if not already started) and sets it to open the artemis home page.



Assumptions

1. Target platform is RedHat Linux.
2. User Guide
 - a. Launching Applications

All external applications are launched via Run->External Tools. If your tool is in the "top 10", it will be directly on the External Tools dropdown menu, otherwise further select "the External Tools..." dialog from the External Tools menu.

 - i. SCP

Possible problems: You may have to klog on your workstation (the one on which Eclipse is running) since the scp launch is made via ssh.

Existing Bugs

1. Problems with SCP running head on Linux
 - a. As of weekend 3-4 Dec 2005, the SCP frequently fails to launch due to a 'Cannot open display' error. Following Ed's change to ASSIGNDIR which permitted this configuration, about 18th Nov, up to weekend 3-4, this configuration worked fine!
 - b. DLOG_TEXT entries have bug: when you hit a button that starts a dialog, the entered text is often rejected because it is apparently invalid. But if you hit the button again, and reenter the same text, that second time the input is accepted. This problem has been tracked to a preceeding NULL in the DLOG input buffer.
 - c. The scp is launched by


```
ssh -X -g -oProtocol=2 ${host} \
@slccom:start_term_run null "scp x5 ${USER}"
```

This successfully opens a DECterm on the user's X11 screen (and thanks to a change made recently by Ed, this works even when the user has sshed to the machine from which they have started the Eclipse iscp session which itself sshed to MCC to start the SCP). However, once that SCP, launched in the DECterm, has exited, one cannot relaunch one. It seems the X window specification expires after one image activation? Not a terribly serious problem, but would be nice if it can be avoided.
 - d. Mouse goes to busy "watch" icon when SCP is not in busy loop.
 - e. Sometimes (for a few hours every few months) the X11 font server used by the xset instruction in the scp.eosi launch config file, is unavailable, which leads to long (2 or 3 minute) lags in scp launch while scp.eosi waits for the xset instruction to timeout. Same problem is presently in the "matlab on pepotics" launch config, since it seems something about matlab on pepotics also uses the font server (though out of control of matlab.eosi).
2. Problem with DECterm as launched by /u/cd/greg/dev/iscp/script/DECterm.eosi command

```
exec ssh -X -oProtocol=2 ${host} "@slccom:remoteterm ${host}"
```

The resulting DECterm does not appear to define a supported CRT - often no problem, but for instance if you subsequently launch LSE it complains "not supported crt". Also the numeric keypad and keys between the QWERTY keys and the numeric keys don't work in the VMS debugger nor in LSE (in /interface=character-cell mode, since /interface=decwindows doesn't work because of the CRT not supported problem). The impact on the VMS debugger makes debugging a scp launched by scp.eosi difficult, since scp.eosi uses slccom:start_term_run, which in turn launches a Decterm like that launched by remoteterm. So basically, remoteterm would be nice, but it doesn't define a working CRT or tty. I get around this by using /afs/slac/g/cd/soft/prod/solaris/bin/DECxterm and executing an ssh within it. Eg, the following creates an xterm which gets around both those problems, but it's an xterm, which VMS users won't be familiar with:

```
/afs/slac/g/cd/soft/prod/solaris/bin/DECxterm -fn
-dec-terminal-medium-*--14-*--*-7-iso8859-*
-xrm "XTerm.VT100.backarrowKey:false"
-bg grey82 -geometry 80x40 -title MCCDEV -e
ssh -t -X -oProtocol=2 mccdev.slac.stanford.edu
```

3. Problem with dm displays from RHEL Linux - some (not all) rapidly updating fields appear to flicker. See, eg bic_lum.dm.
4. jcmlog is very nice, but still has some minor bugs and rough edges:
 - a. No pause and restart. Should be obvious if it's in update mode
 - b. constructed query input is sticky for the first filter dialog (good), but isn't iwhen you subsequently want to modify the filter - you have to start again.
5. At present, all the dm2k displays launched create a new dm2k desktop instance - need to workout how to make >1 display managed by the same instance.

Support Requirements

Hosts on which applications are launched should be known hosts of the machine which AS is running on. Basically this means all users, and mcc nodes, need RSA passwordless authentication to the set of hosts on which things are launched. So we need a table of all hosts that a new user needs authentication on so that sys support can configure each machine in an orderly way.

Need new RH Linux machines that might run AS to have the same configuration for applications. Eg, Firefox must always be in the same place (that is, its executable must always be found at the pathname specified in the launch config), so the launcher config always works. Suggest all additional apps not be put in /scratch but in /apps, or some new Linux fileserver. And there be a written configuration guide maintained by sys group that defines which applications must be present and their precise location.

Firefox extensions required. All new RH machines should be configure with these Firefox extensions:

Java Run Time. If firefox is not installed with JRE, follow instructions at <http://www.mozilla.org/support/firefox/faq#q2.2>, or specifically: cd to your Firefox plugins directory, and create symbolic link to the JRE plugin in AFS (for which you'll ne sudo priv):

```
sudo ln -s /afs/slac.stanford.edu/package/java/@sys/jdk1.4.2/jre/plugin/i386/ns610-gcc32/libjavaplugin_oji.so
libjavaplugin_oji.so
```

Other required extensions:

Adobe PDF viewer

sage RSS browser

Again, if there is a single firefox install on a fileserver, then these must be installed with it.

If AFS is used tokens run out after 25hrs. That means changes to the workspace (Eg displays which are in teh Accelerator Operations set) can't be edited after 25hrs until there's a klog. That'll be a pain.

/nfs/slac/g/esd, or at least the common data directory for applications, must be mounted on all linux machines running the AS. Eg, the ArchiveViewer must be able to find its config files.

Workspace. Only one user may use a given workspace filesystem at a time (that is, a 2nd Eclipse instance will complain if it is asked to use the same workspace directory as an already running Eclipse instance. So, we need a "default" workspace defined in CVS, which users checkout. Then each user uses their own workspace.

Launched Application Suggested Changes for iSCP

SCP

The SCP should be modified so as NOT to include the messages window! jcmlog should be used to view global messages. Some modification may need to be made if some global messages are in fact sent only to an individual SCP and not to the errlog. Local messages should be written by the SCP to sys\$output, and so will be printed in the SCP's Task window by Eclipse.

References and Bibliography

[The Java Developer's Guide to Eclipse](#), 2nd edition. Sherry Shavor et al. Addison-Wesley Professional. 2005.

Proposal for Integrated SLAC Control Program

"iSCP", see attachment iSCP.doc.