
Hands-on: Inca deployment

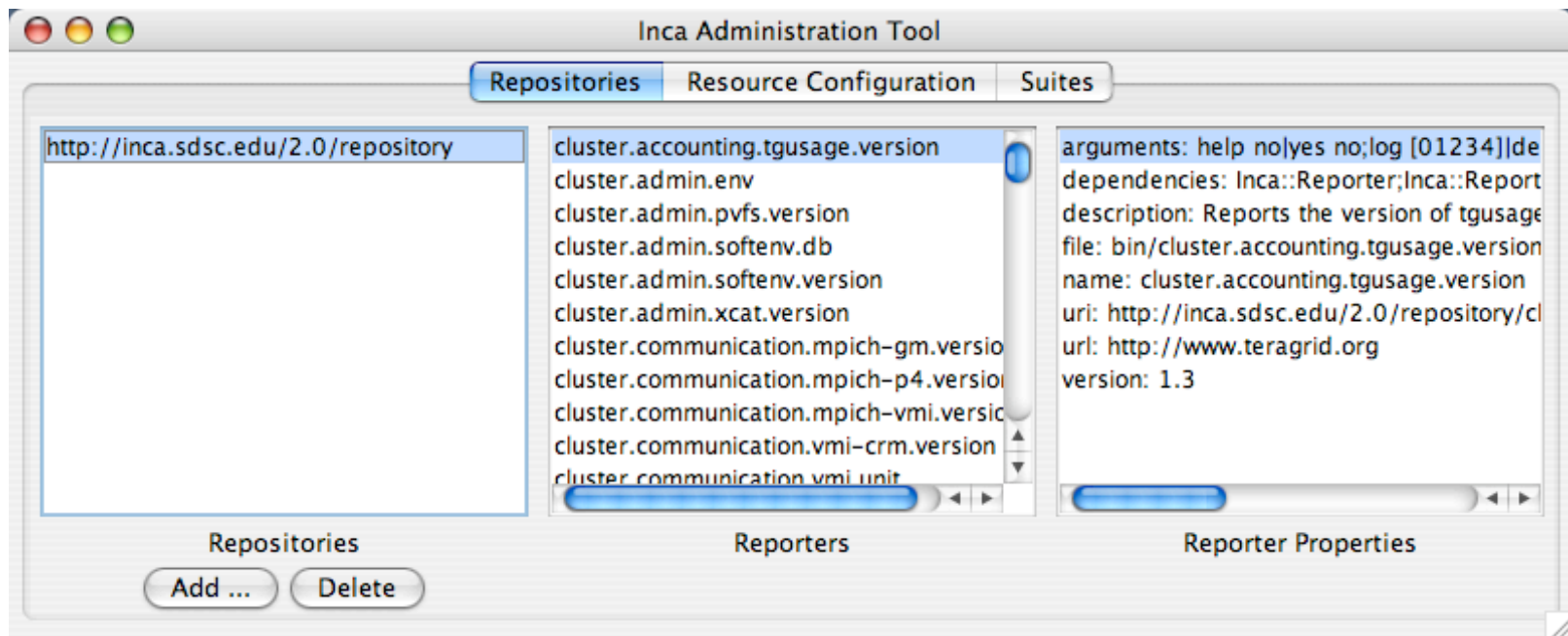
Part 2

Administrative Tasks Covered

1. Stopping/starting Inca components
2. Adding a reporter to a suite
3. Adding a new resource using Globus access method
4. Adding a new suite
5. Adding a new repository
6. Adding/using macros
7. Committing changes
8. Changing the depot backend database to Postgres

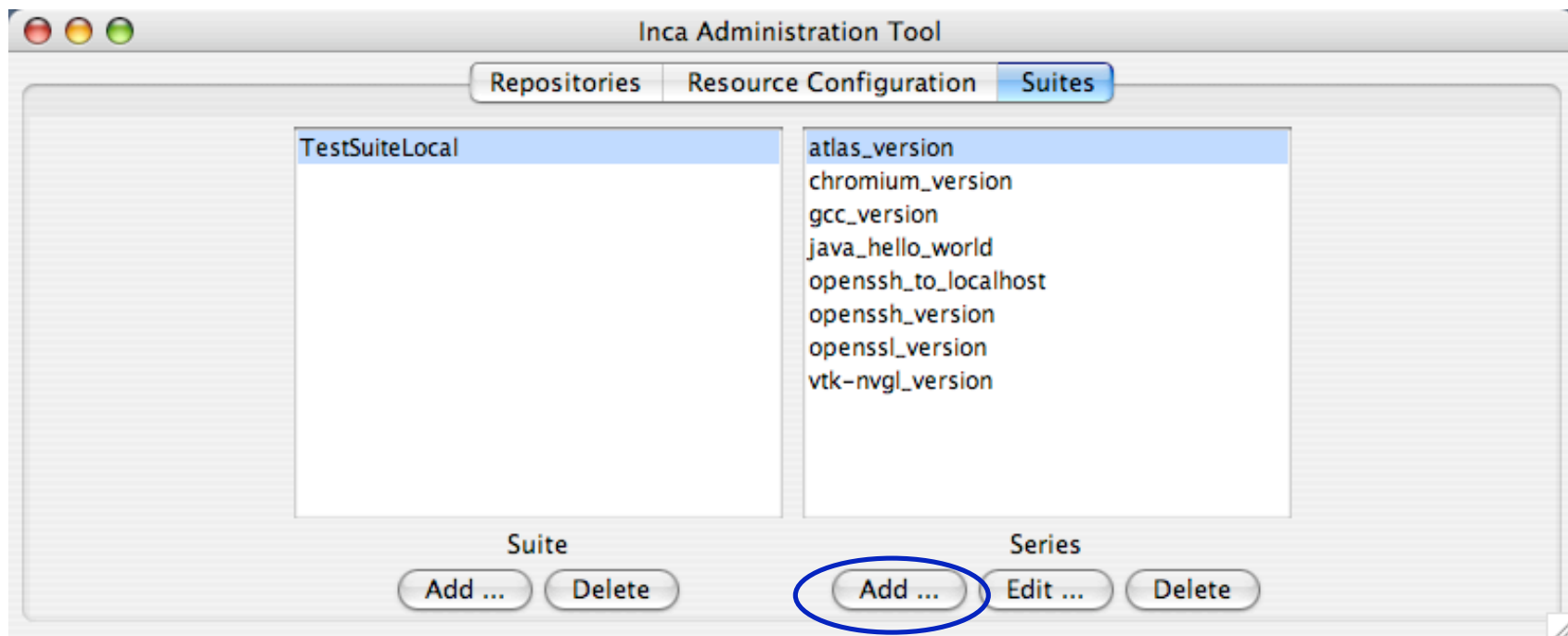
Step 1: Start incat

```
% inca start incat -A localhost:6323 &
```



Step 2: Add a reporter to TestSuiteLocal

a) Click on *Suites* tab



b) Click on *Add* button

Step 2: Add a reporter to TestSuiteLocal (cont.)

- c) Choose the glibc version reporter
- d) Choose *siteA*
- e) Choose an execution time.
This shows every 10 minutes starting at minute 9. Note, if the current time is say 2:00pm, the first time the glibc reporter will be executed is at 2:09pm.

If you want to see the result show up on the web page sooner, set *Starting At* to 2 so that the reporter will run at 2:02pm.

The screenshot shows a configuration window for adding a reporter to TestSuiteLocal. The window has a title bar with standard OS controls. On the left, a list of reporters is shown, with 'cluster.libraries.glibc.version' selected. On the right, a list of resource groups is shown, with 'siteA' selected. Below these lists, the 'Reporter' section is expanded, showing the configuration for 'cluster.libraries.glibc.version'. The configuration includes fields for 'Nickname', 'Exec String', 'Limit CPU (secs)', 'Wall (secs)', 'Memory (MB)', 'Run Every' (set to 10 minutes), 'Starting At' (set to 9), 'Comparator' (set to REComparator), 'Acceptable Output', and 'Notification Email'. The 'Ok' button is circled in blue.

cluster.filesystem.pfs.unit.writeable
cluster.filesystem.scratch.level
cluster.firmware.qlogic.version
cluster.interactive_access.openssh.unit
cluster.interactive_access.openssh.version
cluster.java.sun.version
cluster.libraries.glibc.version
cluster.libraries.glibc.version
cluster.math.atlas.version
cluster.math.blas.unit.level1
cluster.math.blas.unit.level2

defaultvo
siteA

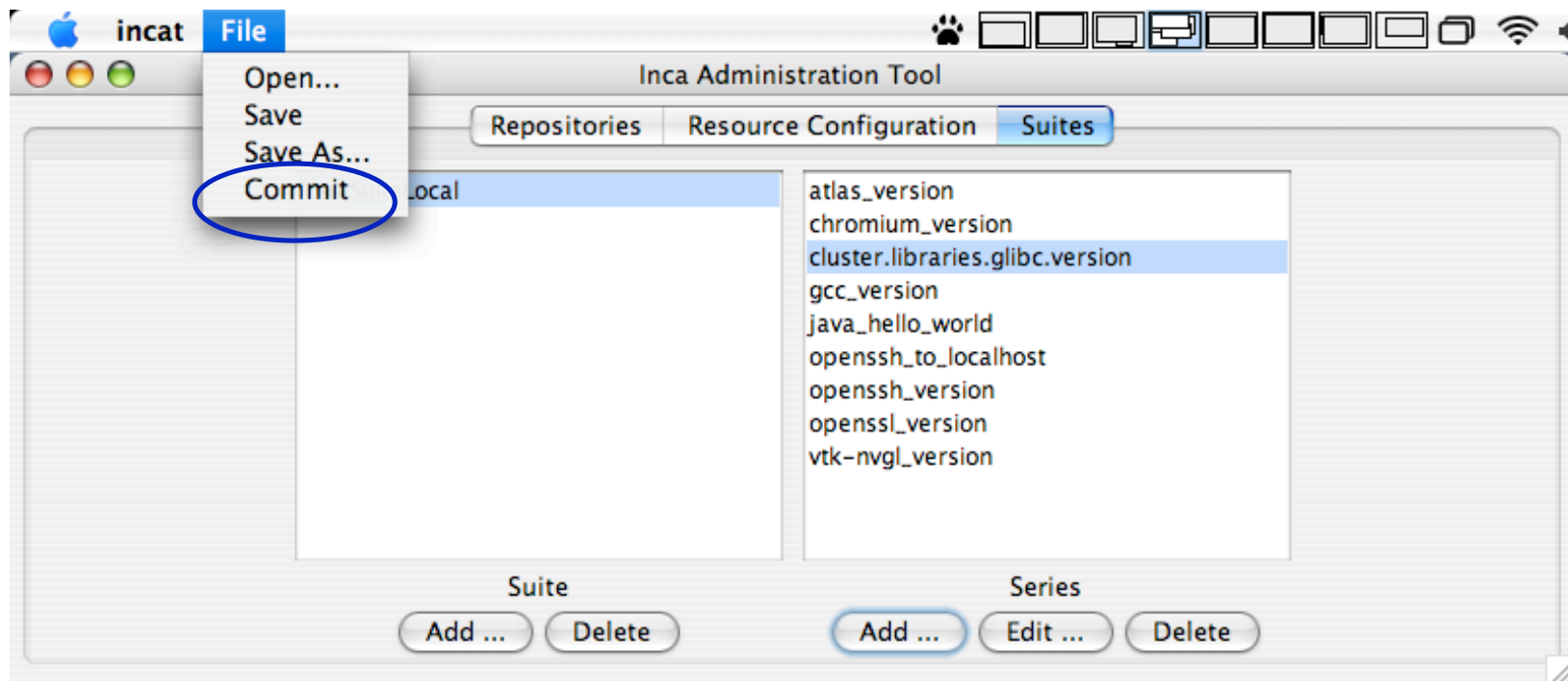
log 0
Arguments

Reporter
(Reports the version of the glibc libraries)

Nickname cluster.libraries.glibc.version
Exec String @EXEC@
Limit CPU (secs) Wall (secs) Memory (MB)
Run Every 10 minutes Starting At 9 9-59/10 ****
Comparator REComparator
Acceptable Output
Notification Email
Cancel Ok

Step 2: Add a reporter to TestSuiteLocal (cont.)

f) Choose *Commit*



Step 2: Add a reporter to TestSuiteLocal (cont.)

- g) Verify result by `swStack.xml` located in `$INCA_INSTALL/etc/common` with <http://inca.sdsc.edu/workshop/swStack.xml>

```
% cd $INCA_LOCATION/etc/common
% rm swStack.xml
% wget \
http://inca.sdsc.edu/workshop/swStack.xml
```

- h) Reload web page and wait for result

Software and Services

COMPILERS	
gcc	siteA
version: any	3.2.3
java	siteA
java_hello_world	fail
GRID SERVICES	
openssl	siteA
version: 0.9.*	0.9.7a
openssh	siteA
version: any	3.9p1
openssh_to_localhost	fail
LIBRARIES	
atlas	siteA
version: any	fail
glibc	siteA
version: any	2.3.2
SCIENTIFIC VISUALIZATION	
vtk-nvgl-gcc	siteA
version: 4.4.2	fail
chromium-gcc	siteA
version: 1.6	fail

Step 3: Modify default configuration

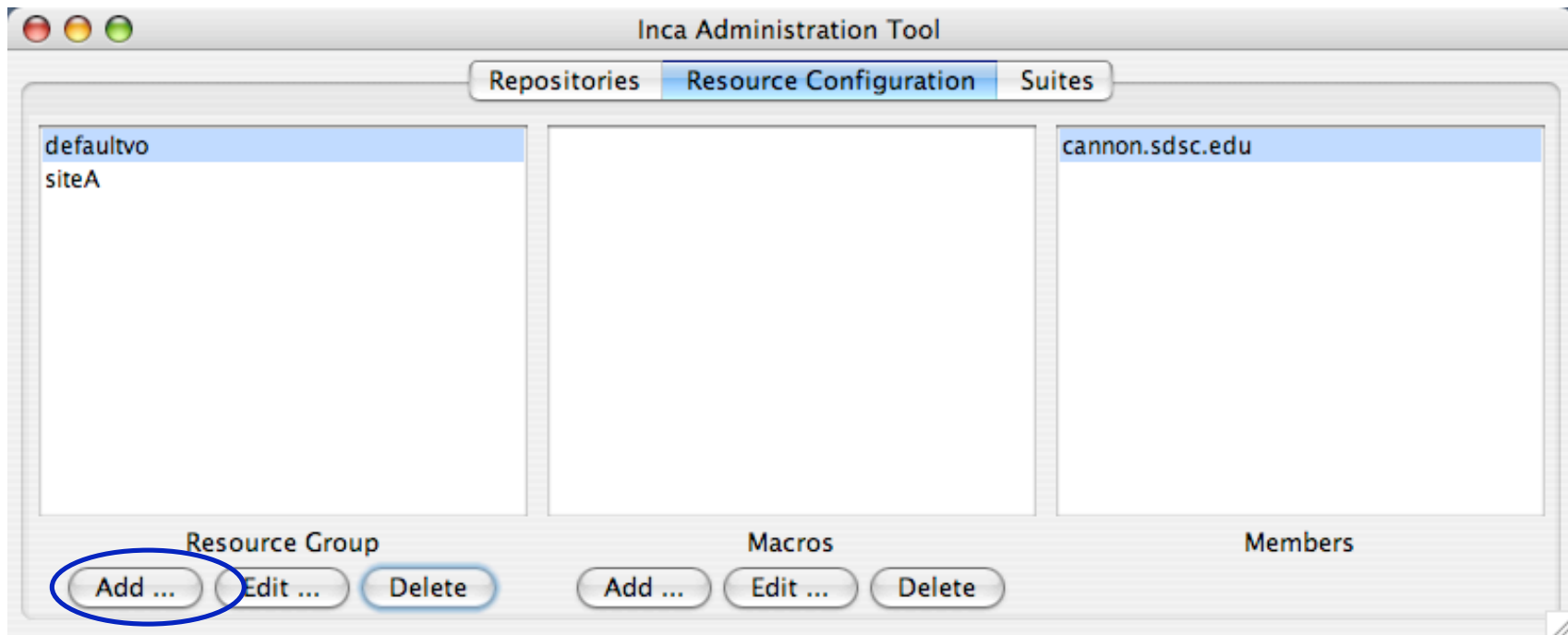
Before adding remote resources, edit `inca.properties` and provide the fqdn of the machine using the following steps:

1. Stop the Inca server components
`% inca stop all`
2. Close incat window.
3. Edit `inca.properties` in `$INCA_INSTALL/etc/common` and replace all instances of `localhost` with the fqdn of the machine you are using (e.g., `hogg.sdsc.edu`).
4. Start the Inca server components
`% inca start all`
5. Start Incat
`% inca start incat -A localhost:6323`

Step 4: Add new resource

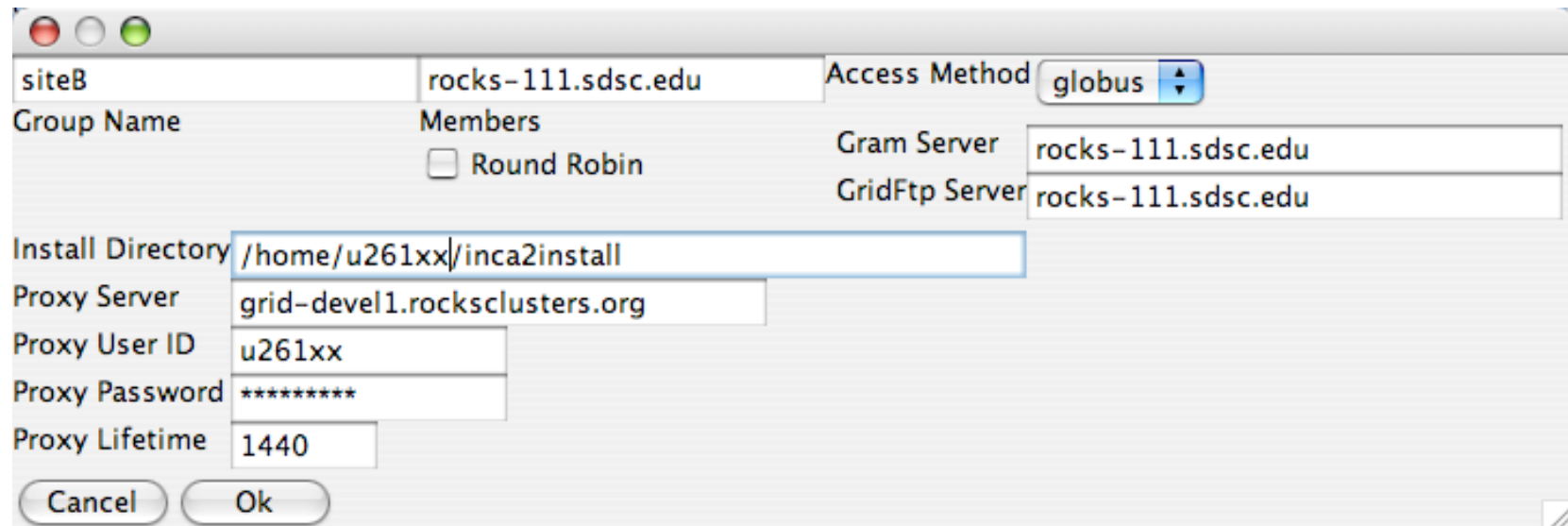
a) Click the *Resource Configuration* tab.

b) Click *Add* under *Resource Group*



Step 4: Add a new resource (cont.)

- c) Add one of the following resources using Globus as the access method called *siteB*:
- rocks-111.sdsc.edu
 - rocks-112.sdsc.edu
 - rocks-113.sdsc.edu
 - rocks-104.sdsc.edu

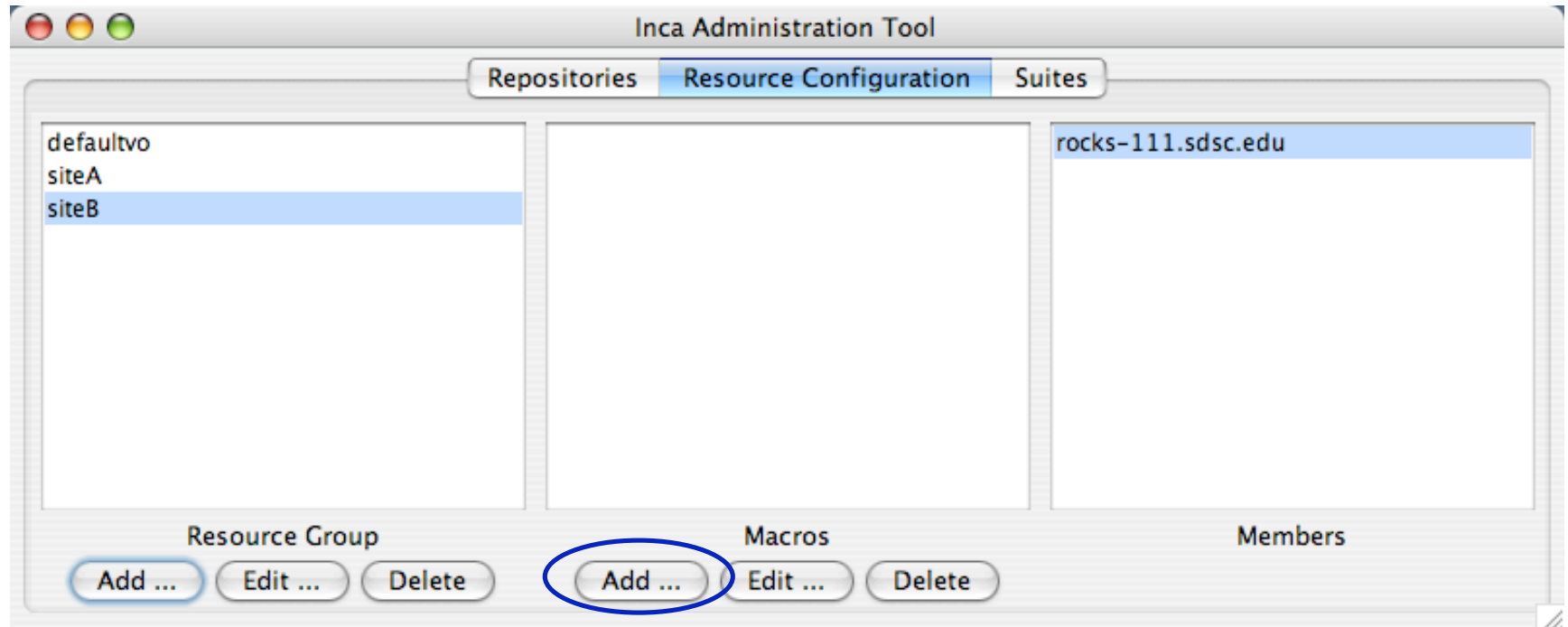


The screenshot shows a configuration window for adding a new resource. The window has a title bar with standard macOS window controls. The main area contains several fields and buttons:

- Group Name:** siteB
- Members:** rocks-111.sdsc.edu
- Access Method:** globus (dropdown menu)
- Round Robin:** ☐
- Gram Server:** rocks-111.sdsc.edu
- GridFtp Server:** rocks-111.sdsc.edu
- Install Directory:** /home/u261xx/inca2install
- Proxy Server:** grid-devel1.rocksclusters.org
- Proxy User ID:** u261xx
- Proxy Password:** *****
- Proxy Lifetime:** 1440
- Buttons:** Cancel, Ok

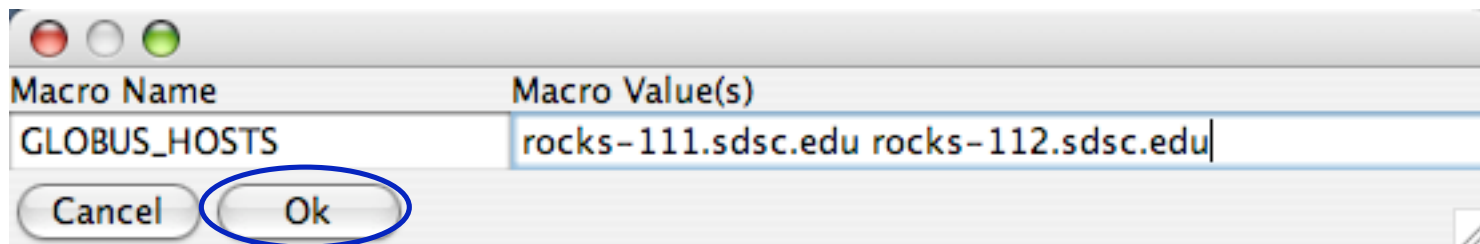
Step 5: Define a resource macro

- a) Highlight *siteB* and click the *Add* button under *Macros*.



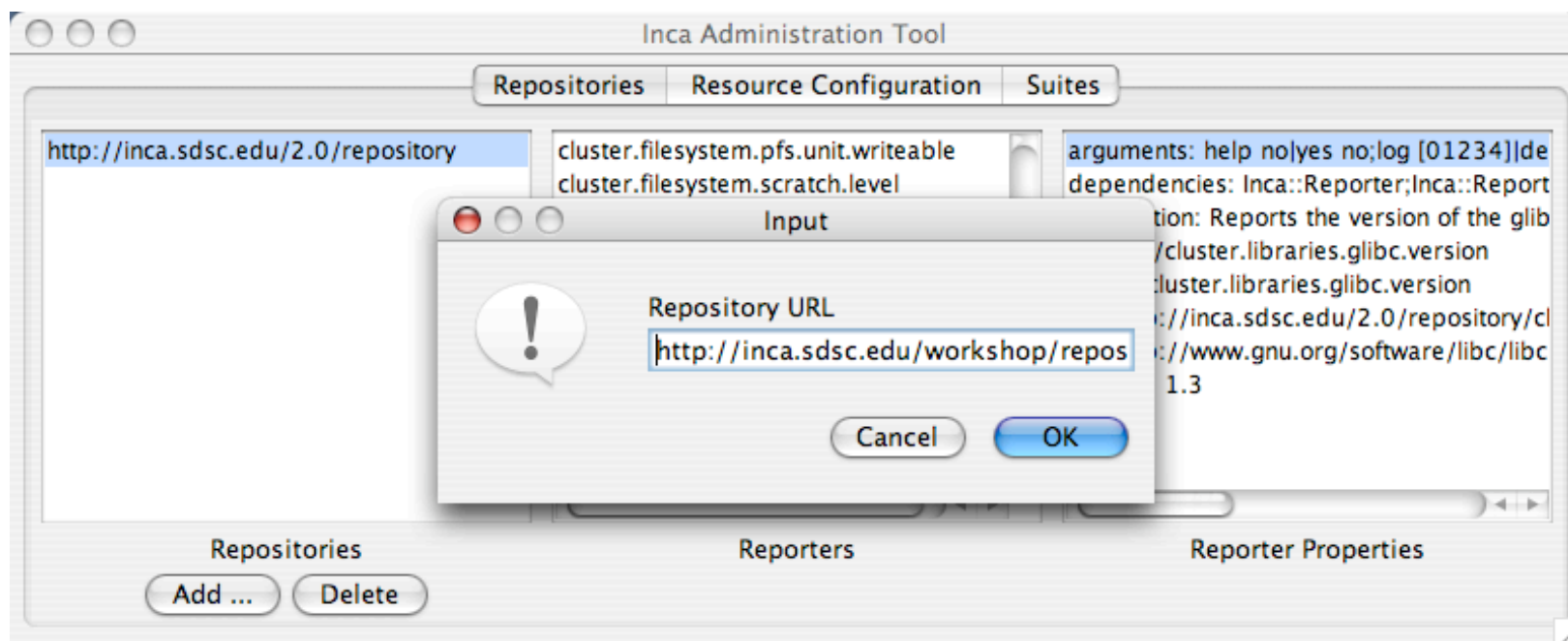
Step 5: Define a resource macro (cont.)

- b) Define the macro *GLOBUS_HOSTS* with values
rocks-111.sdsc.edu rocks-112.sdsc.edu
- c) Click *Ok* button.



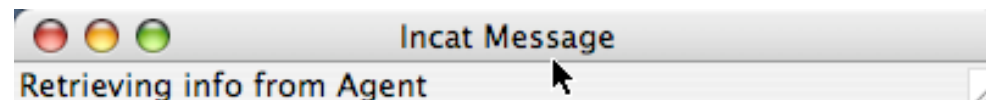
Step 6: Add a new repository

- a) Click on *Repositories* tab
- b) Click the *Add* button under *Repositories* and type in <http://inca.sdsc.edu/workshop/repository>

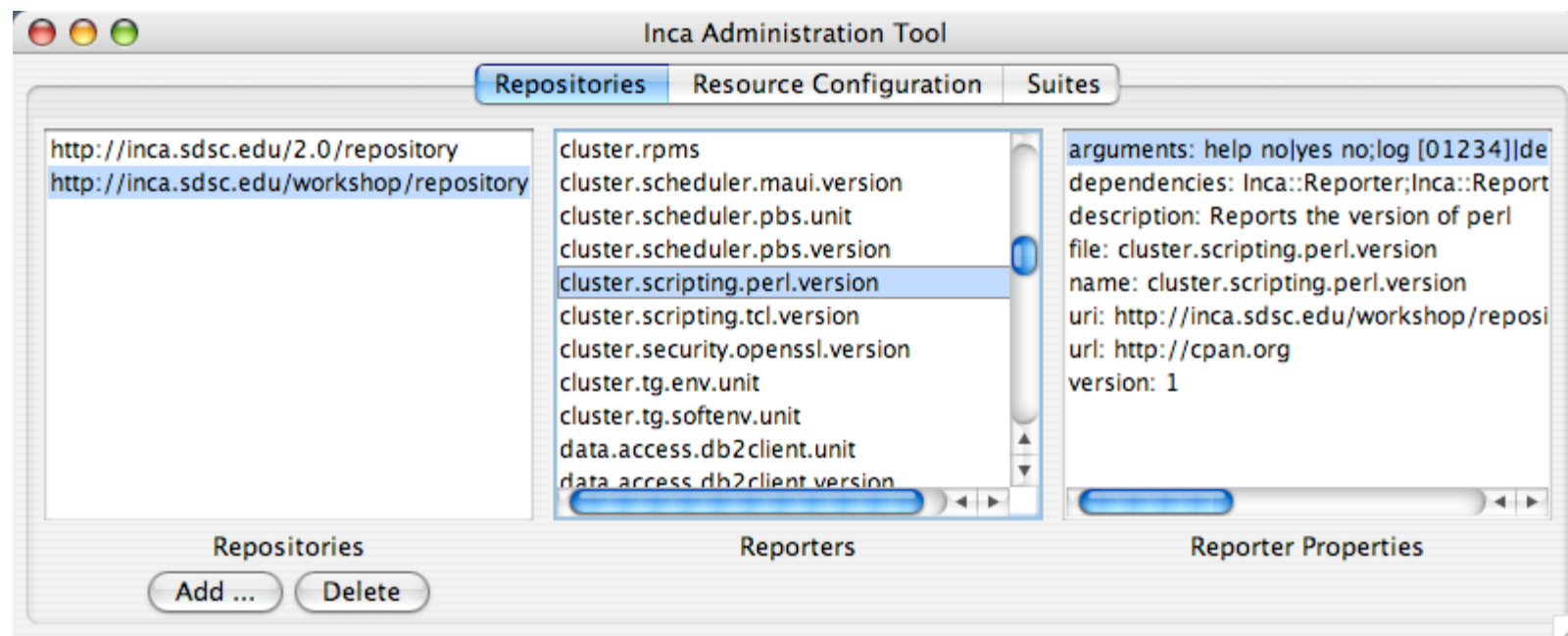


Step 6: Add a new repository (cont.)

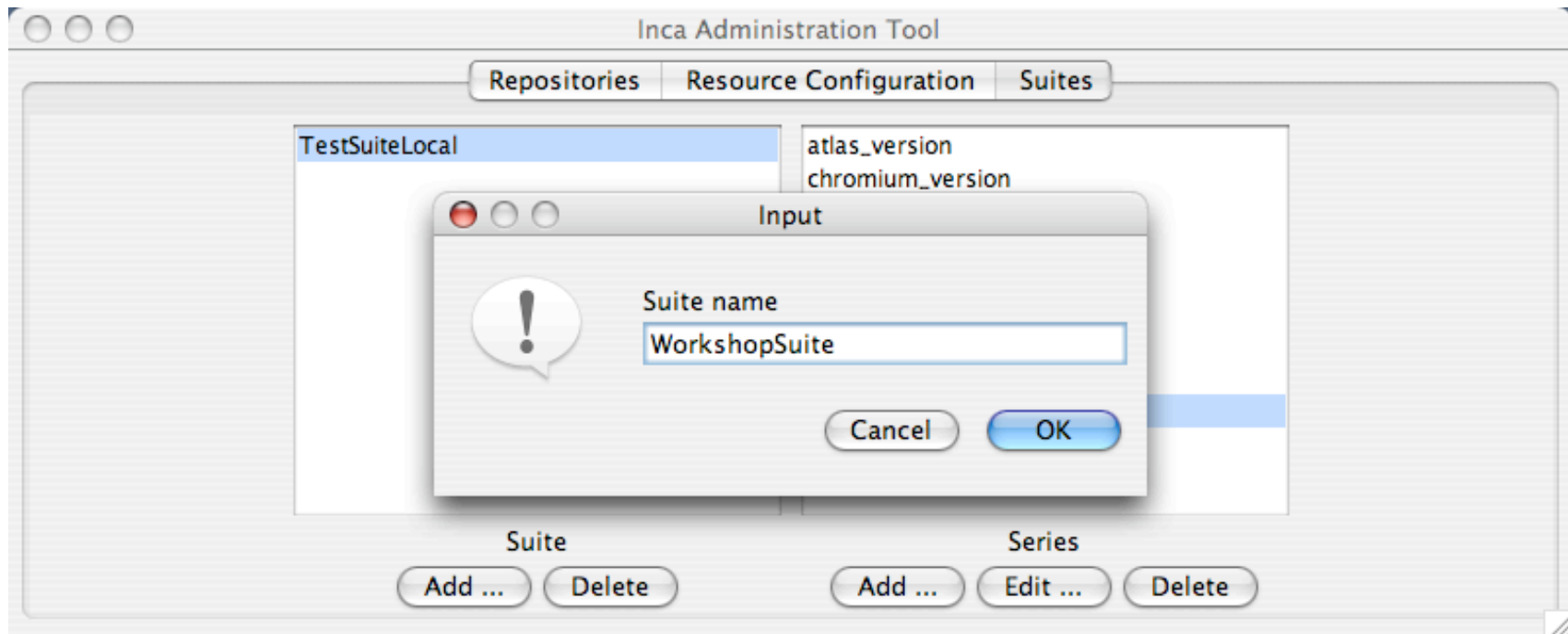
- b) You should see a “Retrieving info from Agent” message.



Once complete, scroll down to see the new reporter:
`cluster.scripting.perl.version`



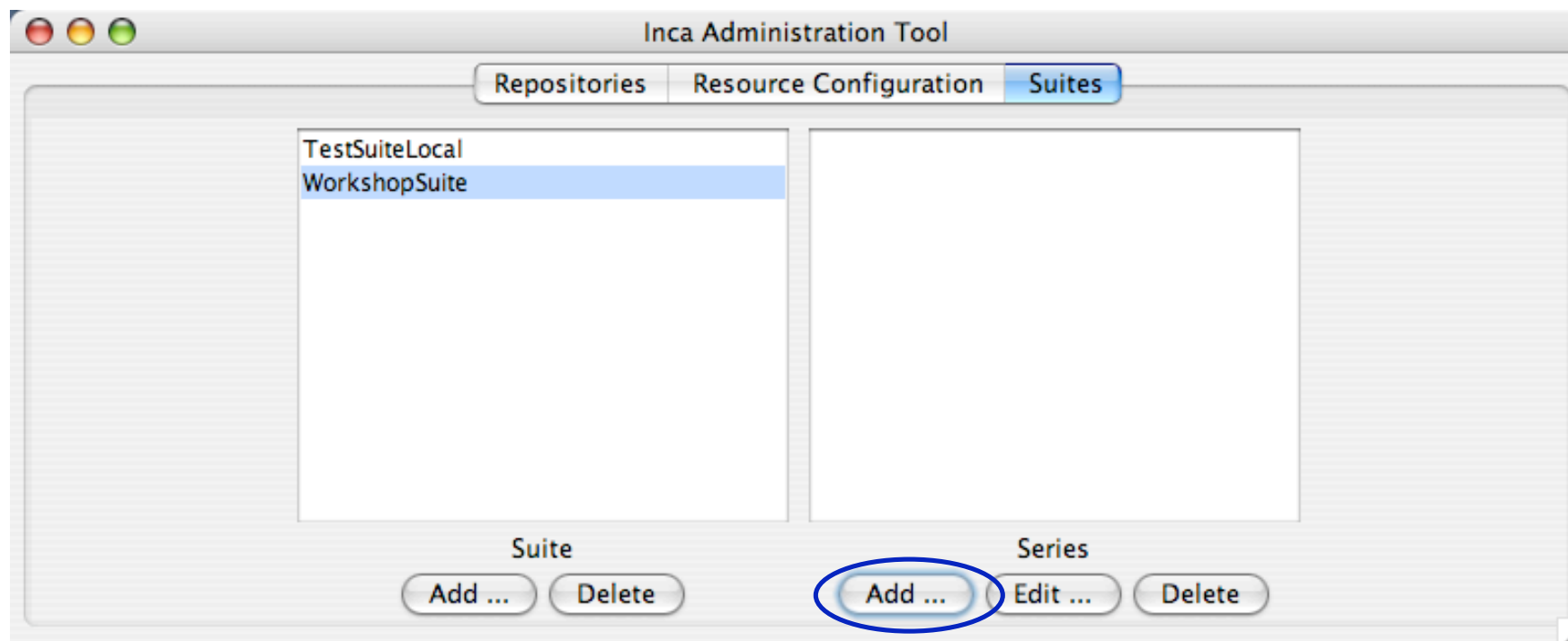
Step 7: Create a new suite



- a) Click on *Suites* tab.
- b) Click the *Add* button under *Suite* and type in *WorkshopSuite* as the name for your new suite

Step 8: Add perl version reporter to suite

a) Click on the *Add* button under *Series*



Step 8: Add perl version reporter to suite (cont.)

- b) Choose perl version reporter
- c) Choose *siteB*.
- d) Choose an execution time. This shows every 5 minutes starting at minute 1.
- e) Click *Ok*.

The screenshot shows the Inca suite configuration window. The 'Reporter' tab is selected, showing a list of reporters on the left and a list of resource groups on the right. The 'cluster.scripting.perl.version' reporter is selected in the left list, and 'siteB' is selected in the right list. Below the lists, the 'Reporter' section is titled '(Reports the version of perl)'. The 'Nickname' field contains 'cluster.scripting.perl.version'. The 'Exec String' field contains '@EXEC@'. The 'Limit CPU (secs)' field is empty. The 'Wall (secs)' field is empty. The 'Memory (MB)' field is empty. The 'Run Every' field is set to '5' minutes. The 'Starting At' field is set to '1'. The 'Comparator' field is set to 'REComparator'. The 'Acceptable Output' field is empty. The 'Notification Email' field is empty. The 'log' field is set to '0'. The 'Arguments' field is empty. The 'Ok' button is circled in blue.

Reporter	Resource Group
cluster.math.blas.unit.level3	defaultvo
cluster.os.kernel.unit.tuning	siteA
cluster.os.kernel.version	siteB
cluster.os.suse.version	
cluster.rpms	
cluster.scheduler.maui.version	
cluster.scheduler.pbs.unit	
cluster.scheduler.pbs.version	
cluster.scripting.perl.version	
cluster.scripting.tcl.version	
cluster.security.openssl.version	

Nickname: cluster.scripting.perl.version
Exec String: @EXEC@
Limit CPU (secs):
Wall (secs):
Memory (MB):
Run Every: 5 minutes Starting At: 1 1-59/5 ****
Comparator: REComparator
Acceptable Output:
Notification Email:
Cancel Ok

Step 9: Add globus gatekeeper test to suite

- Choose globus gatekeeper test reporter
- Choose *siteB*
- Choose *@GLOBUS_HOSTS@* as the host argument.
- Choose *gatekeeper_to_@GLOBUS_HOSTS@* as the *Nickname*.
- Choose */bin/sh -l -c '@EXEC@'* as the Exec string
- Choose an execution time. This shows every 5 minutes starting at minute 3.
- Click *Ok*.

grid.interactive_access.myproxy.version
grid.middleware.globus.unit.date
grid.middleware.globus.unit.env
grid.middleware.globus.unit.gatekeeper
grid.middleware.globus.unit.giis
grid.middleware.globus.unit.gris
grid.middleware.globus.unit.helloworld
grid.middleware.globus.unit.proxy
grid.middleware.globus.version
grid.monitoring.nws.version
grid.parallel_tools.mnrc.unit

defaultvo
siteA
siteB

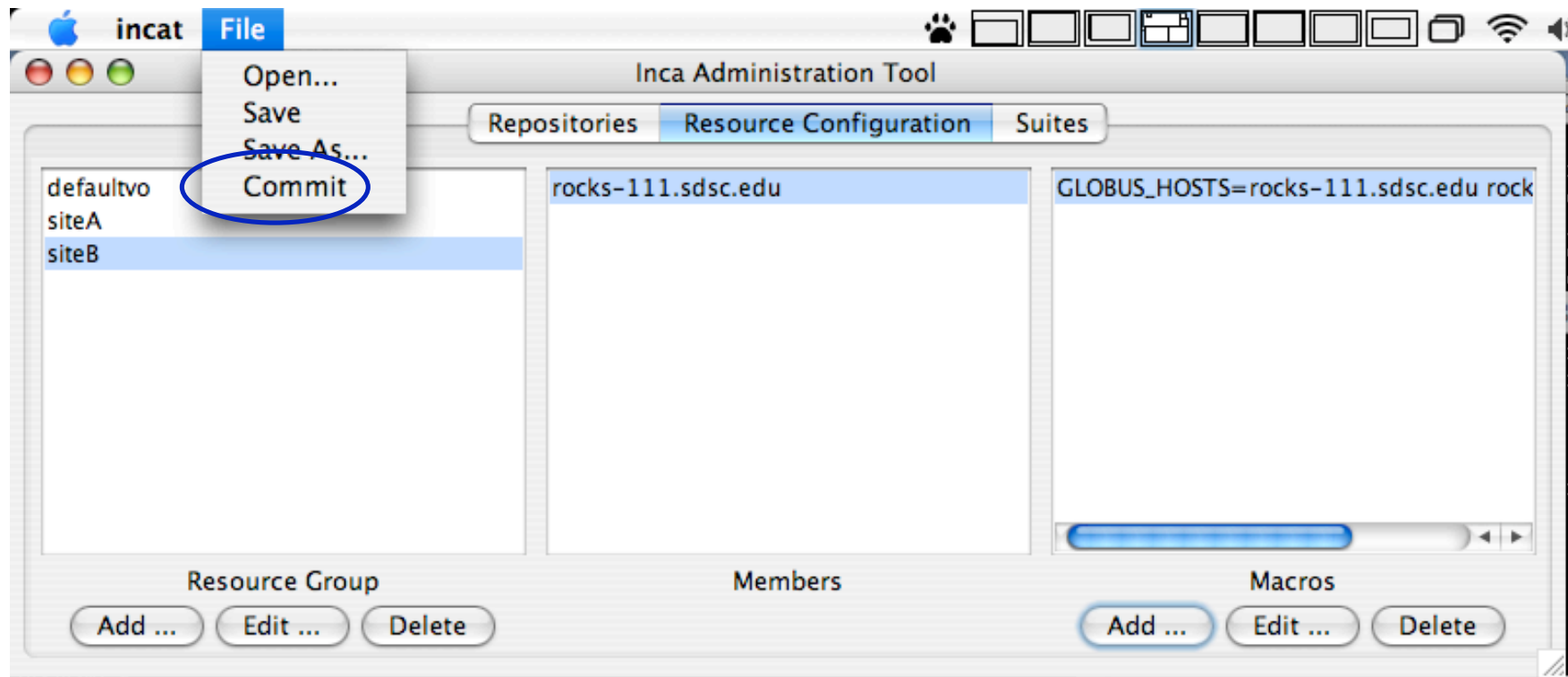
host @GLOBUS_HOSTS@
log 0
Arguments

Reporter
(Checks that the gatekeeper at a host is accessible from the local machine)

Nickname gatekeeper_to_@GLOBUS_HOSTS@
Exec String /bin/sh -l -c '@EXEC@'
Limit CPU (secs) Wall (secs) Memory (MB)
Run Every 5 minutes Starting At 3 3-59/5 ****
Comparator REComparator
Acceptable Output
Notification Email
Cancel Ok

Step 10: Commit changes

Choose *Commit*



Step 11: View result

- a) Copy swStack2.xml from
<http://inca.sdsc.edu/workshop/swStack2.xml>
into
\$INCA_INSTALL/etc/common
% cd
\$INCA_INSTALL/etc/common
% wget \
<http://inca.sdsc.edu/workshop/swStack2.xml>

- b) Open page:
<http://host.sdsc.edu:8080/xslt.jsp?xsl=http://host.sdsc.edu:8080/swStack.xsl&resourceID=siteB&suiteName=WorkshopSuite&xmlFile=swStack2.xml>

Software and Services

GRID SERVICES	
globus	siteB
gatekeeper_to_rocks-111.sdsc.edu	pass
gatekeeper_to_rocks-112.sdsc.edu	pass
LANGUAGES	
perl	siteB
version: any	5.8.5

Step 12: Use postgres as db backend

- a) Stop the depot
`% inca stop depot`
- b) Edit `$INCA_INSTALL/etc/hibernate.properties`
 - i. Comment out the first 5 lines which specifies for hibernate to use hsql as its backend database:

```
#hibernate.dialect=org.hibernate.dialect.HSQLDialect
#hibernate.connection.driver_class=org.hsqldb.jdbcDriver
#hibernate.connection.url=jdbc:hsqldb:test
#hibernate.connection.username=sa
#hibernate.connection.password=
```

Step 12: Use postgres as db backend (cont.)

- ii. Uncomment the second block of lines 8-13 which specifies for hibernate to use postgres:

```
# Settings for a local PostgreSQL database.  
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect  
hibernate.query.substitutions=yes 'Y', no 'N'  
hibernate.connection.driver_class=org.postgresql.Driver  
hibernate.connection.url=jdbc:postgresql:test  
hibernate.connection.username=test  
hibernate.connection.password=
```

Step 12: Use postgres as db backend (cont.)

- iii. Replace the value `jdbc:postgresql:test` for property `hibernate.connection.url` on line 11 with the following:

```
hibernate.connection.url=  
jdbc:postgresql://rocks-111.sdsc.edu/u261xx
```

where `u261xx` is your username.

- iv. Replace the value “test” for property `hibernate.connection.username` on line 12 with your username

```
hibernate.connection.username=u261xx
```

Step 12: Use postgres as db backend (cont.)

- c) Download the postgres jdbc jar file `pg74.216.jdbc3.jar` to your `$INCA_INSTALL/lib` directory

```
% cd $INCA_INSTALL/lib
```

```
% wget \
```

```
http://inca.sdsc.edu/workshop/pg74.216.jdbc3.jar
```

- d) Initialize the depot (set up the Inca tables)

```
inca depot -d
```

You should see something like the following message:

```
Initializing c3p0 pool...
```

```
...
```

```
Database Initialization Completed
```

Step 12: Use postgres as db backend (cont.)

- d) Start the depot
`% inca start depot`
- e) Reload your web page from Step 2 or Step 11. The results will be n/a (because you are using a fresh database that has not been configured to run any reporters)

Software and Services

GRID SERVICES	
globus	siteB
gatekeeper_to_rocks-111.sdsc.edu	n/a
gatekeeper_to_rocks-112.sdsc.edu	n/a
LANGUAGES	
perl	siteB
version: any	n/a

Step 12: Use postgres as db backend (cont.)

- f) Use incat to delete WorkshopSuite. Re-add the WorkshopSuite with the perl version reporter as shown in Step 7 & 8. Wait for web page to update.

Software and Services

GRID SERVICES	
globus	siteB
gatekeeper_to_rocks-111.sdsc.edu	n/a
gatekeeper_to_rocks-112.sdsc.edu	n/a
LANGUAGES	
perl	siteB
version: any	5.8.5