Readme

Platform Computing

Contents

1

Release Notes

What's new in Platform Application Center 8.0.1

Supported operating systems

Platform Application Center is now supported on Linux 2.6 glibc 2.3 x86 64 bit, Red Hat 6.

Application templates

The following application templates are now included: NWCHEM, HMMER, ClustalW, CMGL_GEM, CMGL_IMEX, CMGL_STARS.

Easier upgrade with RPM

It is now easier to upgrade. You can use `rpm -Uvh` *filename*.`rpm` to upgrade Platform Application Center.

Jobs page now includes jobs, job arrays, job flows, and job groups

You can now view jobs, job arrays, job flows, and job groups from one page, the Jobs page by selecting Jobs > By State.

Use the Type column to sort the list, or Options > Filters to refine the data you view. The navigation tree now also includes states.

Alias displayed for the job directory instead of path in Job Data

When you define an alias for the job directory with `<pac:spooler-info>`, the alias is displayed in the Job Data page instead of the job directory path. You can see this in:

- Job Data > By Job, Job Directory column
- Job Data > By Flow, Job Flow Directory column

Custom page creation

- You can now add or remove pages in `services.xml` and your changes are automatically recognized by Platform Application Center when you click Refresh. You no longer need to restart Plaform Application Center.
- Call an action from another action by using `<pac:call>` in `<pac:action>`.
- Redirect to the Flow Details page with `<pac:jobflow-submit>`.
- Redirect to the Job Details page with `<pac:job-submit>`.
- Redirect to the Job Group Details page with `<pac:jobgroup-submit>`.
- Redirect to the job directory with `<pac:go-to-spooler>`.

Monitor and control jobs in LSF job groups

You can now submit, view and control jobs in job groups in Platform Application Center. To submit jobs, you need to change the application templates and add a text field or a new dropdown field. To monitor jobs in job groups, go to Jobs > By State.

Resubmit button in Job Details page

There is a new action in the Job Details page: Resubmit.

When you click Resubmit, the job submission form is displayed pre-filled with the parameters that were used to submit the job from which the button was clicked. You can then change any parameters and click Submit to resubmit the job with the new parameters.

Enhanced log levels

There are now clearly defined log levels in the Platform Application Center log: DEBUG, INFO, WARNING, ERROR, FATAL. By default, the log level is INFO.

Retrieve the external status for a job with Web Services

You can use web services to retrieve the external status for a job.

The output of `pacclient.py job` now has a new column: BSTATUS. This column displays the external status for a job.

The following functions have been updated in the Python API to return the external status of a job:

- `getJobInfo(jobId)`
- `getJobForStatus(jobStatus)`
- `getJobForName(jobName)`

The output on success of these functions now includes `<extstatus>`.

You can also view this status in the Jobs page, with the Ext Status column. You will need to select the column for display in Options > Preferences.

View license usage through Platform Application Center

Platform Application Center now displays license usage when Platform License Scheduler 8.0 is installed and licensed. Platform Application Center displays Resources > Licenses. You can view license usage by feature, by job, by project, and by cluster.

Requirements:

- Platform License Scheduler 8.0 is installed in your LSF cluster.
- You have an enterprise license for Platform Application Center.
- Platform Application Center 8.0 is installed and can access $LSF_ENVDIR.

Installation:

After installation, you need to enable viewing licenses through Platform Application Center:

1. In `$GUI_CONFDIR/pmc.conf`, set ENABLE_LS_GUI =Y.
2. Restart the web server.

```
pmcadmin stop
perfadmin stop all
pmcadmin start
perfadmin start all
```

Platform Analytics now embedded in Platform Application Center

Platform Application Center now embeds Platform Analytics 8.0.

Requirements:

- Platform Analytics 8.0 is installed in your LSF cluster.
- You have an enterprise license for Platform Application Center.
- Platform Application Center 8.0 is installed and can access $LSF_ENVDIR.
- The Vertica client driver is installed on your Tableau server. You can download the driver package from the platform FTP site, in the same location as Platform Analytics.
- You have installed the Platform Application Center Analytics add-on package. The package comes with installation instructions. You can download the add-on package from the Platform FTP site, in the same location as Platform Analytics.

Installation:

- After installation, you need to enable viewing Platform Analytics reports through Platform Application Center:

  1. In $GUI_CONFDIR/pmc.conf, set ENABLE_PA_REPORTS=Y.
  2. Restart the web server.

     ```
     pmcadmin stop
     perfadmin stop all
     pmcadmin start
     perfadmin start all
     ```

Access to reports:

- When the add-on has been installed, the Reports in Application Center are replaced with the reports from Platform Analytics. You can access them through Reports > By Workbook.

Users and access control:

- In order to have access to the reports in Platform Application Center, users must be a Viewer, Interactor, or administrator in Tableau. Users can use the same UNIX account name as they use to log on to Tableau.
- Through the Platform Application Center Settings tab, the Cluster administrator can configure access per workbook. Users will only be able to see workbooks for which they have permission in Platform Application Center or in Tableau.

User roles:

Role	Permissions
Report administrator	Platform Application Center automatically loads Tableau Licensed users with the new role Report administrator. A user with the Report administrator role can: • Display the report list and view a report • View past reports • Manage report scheduling • Subscribe to a report, or unsubscribe from a report to receive emails when reports are updated • Add extra email addresses for sending reports
Normal user	A user with the Normal user role can: • View past reports • Subscribe to a report, or unsubscribe from a report to receive emails when reports are updated
Cluster administrator	A user with the Cluster administrator role can: • Configure permissions on who can view a workbook through Platform Application Center's access control • Add emails to user properties for application users through **Settings** > **Users and User Groups List**.

System Requirements

This section describes system requirements for installing and running Platform Application Center.

Supported operating systems

- Linux 2.6 glibc 2.3 x86 64 bit, Red Hat 5+ and 6
- Linux 2.6 glibc 2.3 x86 64 bits, SUSE 10.x, SUSE 11

Supported authentication

- Password File
- NIS
- LDAP
- AD (Active Directory)

Supported web browsers

- Internet Explorer 7 and 8 on Windows
- Firefox 3.5 and 3.6 on Windows

Adobe Flash version 8 or later is required to access report charts.

Supported databases

- MySQL 5.x
- Oracle 9i Enterprise Edition for production clusters
- Oracle 10g Enterprise Edition for production clusters

Important:

The Apache Derby database is no longer supported in Platform Application Center.

Supported Java Runtime Environment (JRE)

- JRE 1.5 or higher

Supported LSF versions

- Platform LSF 7 Update 3 or higher
- Platform LSF 8
- Platform LSF 8.0.1

Disk requirements

- Disk Size:

  Each host should have at least 350 MB of local disk space to install. To support ongoing logging of important data, we recommend 2 GB or more.
- Job Data:

Make sure you have enough space for job data. By default, all job data is stored under the directory /home/.

Web server host

For best performance, do not use the master host as the web server host.

- If EGO is enabled, the web server must be one of the LSF master candidates.
- If EGO is disabled, any LSF server host can be the web server host as long as the host can access (read and write) the following:

$LSF_ENVDIR

$LSF_SERVERDIR

$LSF_WORKDIR

$LSF_LOGDIR

International language support

Platform Application Center displays the language preference set in your web browser. If you set a new language preference or change the order of your preferred languages in your browser options, you must log out and log back in to Platform Application Center to make the change take effect.

Besides the default English (en), the following browser language settings are supported:

- French (fr-FR). Some messages have not been translated. The properties file will need to be updated.
- Japanese (ja and ja-JP)
- Chinese (zh-CN)

See your browser help for more information about setting language preferences.

Tested applications

The following versions of supported applications have been tested with Platform Application Center:

Note:

These are tested application versions. Job submission forms can be customized to support other versions.

- FLUENT

    - 12.1.2
    - 12.0.16
    - 6.3.26
- ANSYS

    - V121
    - V120
    - V110
    - V100
- LS-DYNA

    - 971
    - 970

- ABAQUS
  - 6.9-1
- NASTRAN
  - MSC 2008
- ECLIPSE
  - 2009.1
  - 2010
- CFX
  - CFX-5 (Ansys Suite)
- BLAST
  - 2.2.20
- CLUSTALW
  - 2.1
- HMMER
  - 3.0
- NWCHEM
  - 6.0
- CMGL_GEM
  - 2008.12
  - 2009.13
- CMGL_IMEX
  - 2008.11
  - 2009.11
- CMGL_STARS
  - 2008.12
  - 2009.11

Known Issues and Limitations

License file cannot include Platform Application Center 1.3 and 8.0

It is not supported to have both the Platform Application Center 1.3 license keys and the version 8.0 license keys in the same file. You cannot run both Platform Application Center 1.3 and 8.0 in parallel on the same hosts.

/var/logs/messages has multiple lines added for each action in Platform Application Center

For each user request, an action will be logged in the oeprating system /var/logs/messages file. This causes the log file to increase in size very quickly.

The workaround is to disable the system log for /bin/sh -c.

Minimum RHEL5.3 and 5.5 missing compat-libstdc++-33-3.2.3-61

The compat-libstdc++-33-3.2.3-61 may be missing from the minimum RHEL 5.3 and 5.5 installation. Platform Application Center requires this package. Run the following commmands to make sure that the package is installed correctly:

1. **pmcadmin stop**
2. **yum install compat-libstdc++-33-3.2.3-61**
3. **pmcadmin start**

File list display limitation

If a folder in a file list contains more than 4000 files or folders, Platform Application Center can only can display a maximum of 4000 files in a file selection list.

Special characters in file names

Special characters single quote ('), double quote (") and dollar sign (S) are not supported in file names. File management and file selection in Platform Application Center will not work properly for files with these characters in the file name.

Platform Application Center host cannot have underscores in the host name

This is a known issue in Internet Explorer and Firefox.

On Internet Explorer, setting cookies fails when the host name of the server contains underscores (_). This issue prevents users from logging in to Platform Application Center. On Firefox, this issue causes downloading and uploading files to fail.

To work around the problem, change the host name to remove underscores or use the IP address to access the Platform Application Center server.

Per-job remote Console is supported for shared disk installation only

For per-job remote consoles, Platform Application Center creates a VNC session file in $GUI_WORKDIR, which must be located a shared location (for example, /opt/pac). Platform Application Center cannot create a VNC session on compute nodes if $GUI_WORKDIR is a local directory.

You must install Platform Application Center in a shared location enable failover to use the per-job remote consoles.

LSF API socket in use error with more than 15 users

When more than 15 users are logged in accessing application submission forms, monitoring jobs, etc., the LSF API gives socket in use errors.

To work around the problem, the administrator must set the parameter LSF_NON_PRIVILEGED_PORTS=Y in lsf.conf.

Linux run level 5 required for RGS jobs

LSF RGS jobs require run level 5 in order to use startx to start the RGS sender process. If the run level is not 5, the X-Windows server cannot start. Make sure the Linux run level on your system is set to 5 (the default may be 3 or some other value on your system).

1. Edit /etc/inittab and modify the following:

```
# Default runlevel. The runlevels used by RHS are:
# 0-halt (Do NOT set initdefault to this)
# 1-Single user mode
# 2-Multiuser, without NFS (The same as 3, if you do not have networking)
# 3-Full multiuser mode
# 4-unused
# 5-X11
# 6-reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

2. Reboot the machine to make the change take effect, or log on as root and start the X Server manually:

```
/sbin/init 5
```

Job notification does not record status change from Exited to Pending

When job status changes from Exited to Pending the status change is not updated.

For example, if you requeue an Exited job, status changes to Pending in LSF but the change is not displayed. When the job starts running, the status notification message changes to "Exited to Running". The Exited to Pending change is not shown.

Cannot connect to server when opening a remote console

When a user tries to open a remote console through Remote Consoles > Open my console, a VNC console window is displayed, but cannot connect to the server. No VNC session is created on the master host.

To work around the problem, check $GUI_WORKDIR/.vnc directory. Remove any ${user} subdirectory for the user having a problem connecting. This will remove the VNC session file created in $GUI_WORKDIR/.vnc/${user}, and the user should be able to connect successfully.

Job Flows Global View display limitation

In the Job Flows > Global View, there is a limit to the number of dependencies that can be displayed. If the number of dependencies is more than 48, the name will be missing on some items in the view. The issue appears only on Firefox 3.6 with Shockwave 10 Flash plugin when the number of dependencies in the view is greater than 48.

Process Manager job templates are not supported

Platform Process Manager job templates are not supported in Platform Application Center. Use Platform Application Center application templates for job for submissions to job flows.

Access denied error when triggering job flow from a custom page

When triggering a job flow from a newly created custom submission form, an access denied error is displayed.

To avoid this problem, you must restart Platform Application Center before using a new custom page added through a service in `services.xml`. New services are automatically synchronized with Platform Application Center access controls every host. To access a new custom page immediately, you must log out and log back in to Platform Application Center to have the new service to be added into the access controls.

Browser error when switching to HTTPS

After enabling HTTPS using `pmcadmin https enable`, Firefox and Internet Explorer complain that Platform Application Center uses an invalid untrusted security certificate.

Platform Computing provides a self-signed security certificate for Platform Application Center. Because it is self-signed, the certificate is not trusted by the browser. When you get this error, you can tell your browser to treat the self-signed certificate as exception in browser security.

On Internet Explorer, you see the error "Certificate Invalid":

1. Click "View certificates".
2. Click "Continue to this website (not recommended)."

On Firefox, you see the error "This Connection is Untrusted":

1. Under "What Should I Do?", click "I Understand the Risks".
2. Click Add Exception to tell Firefox to start trusting the Platform certificate.
3. On the Add Security Exception dialog, click Confirm Security Exception.

Get Technical Support

Contact Platform Computing

Contact Platform Computing or your Platform Application Center vendor for technical support. Use one of the following to contact Platform technical support:

Email

support@platform.com

World Wide Web

www.platform.com

Mail

Platform Support
Platform Computing Inc.
3760 14th Avenue Markham
Ontario Canada L3R 3T7

When contacting Platform, please include the full name of your company.

See the Platform Web site at *www.platform.com/company/contact-us* for other contact information.

Get patch updates and other notifications

To get periodic patch update information, critical bug notification, and general support notification from Platform Support, contact *supportnotice-request@platform.com* with the subject line containing the word "subscribe".

To get security related issue notification from Platform Support, contact *securenotice-request@platform.com* with the subject line containing the word "subscribe".

Help and documentation

To get started using Platform Application Center, click Help in the Platform Application Center console.

To install Platform Application Center for the first time, see *Installing Platform Application Center*.

To upgrade from Platform Application Center version 1.3, see *Installing Platform Application Center*.

To configure and administer Platform Application Center, see *Administering Platform Application Center*.

Learn More

Information about Platform products is available from the following sources:

- World Wide Web and FTP
- Platform documentation
- Platform training

World Wide Web and FTP

Information about Platform products is available from the Platform FTP site (`ftp.platform.com`) or the Platform Web site at *www.platform.com* .

If you have problems accessing the Platform web site or the Platform FTP site, send email to *support@platform.com*.

my.platform.com

`my.platform.com`—Your one-stop-shop for information, forums, e-support, documentation and release information. `my.platform.com` provides a single source of information and access to new products and releases from Platform Computing.

On the Platform Application Center product page of `my.platform.com`, you can download software, patches, updates and documentation. See what's new, check the system requirements.

Platform training

Platform's Professional Services training courses can help you gain the skills necessary to effectively install, configure and manage your Platform products. Courses are available for both new and experienced users and administrators at our corporate headquarters and Platform locations worldwide.

Customized on-site course delivery is also available.

Find out more about Platform Training at *www.platform.com/services/training*, or contact *Training@platform.com* for details.

2

New Feature Documentation

Upgrade Platform Application Center 8.0 to 8.0.1

To upgrade Platform Application Center from version 8.0 to 8.0.1, you will need to back up your existing installation, shut down Platform Application Center, apply the update with RPM, update your database schema, then start Platform Application Center .

Install the upgrade

You must have installed Platform Application Center 8.0.

Download Platform Application Center 8.0.1. You do not need a new license file.

The following steps assume Platform Application Center is installed in /opt/pac. Replace with your actual installation directory.

If you did not install in /opt/pac, you will need to use --prefix when running RPM to specify your desired directory.

1. As root, log on to your Platform Application Center 8.0 host.
2. Set your Platform Application Center environment:

   For example:

   - For csh or tcsh:

     ```
     %  source /opt/pac/cshrc.platform
     ```
   - For sh, ksh, or bash:

     ```
     $  . /opt/pac/profile.platform
     ```
3. Stop Platform Application Center services.

   ```
   pmcadmin stop
   ```
   ```
   perfadmin stop all
   ```
4. Check services have stopped.

   ```
   pmcadmin list
   ```
   ```
   perfadmin list
   ```

   You should be able to see the WEBGUI, jobdt, plc, and purger services stopped.
5. Make a copy of your Platform Application Center 8.0 directory.

   For example, /opt/pac.

   This is so that if you ever want to roll back, you can.
6. Delete your temporary directory.

   For example:

   ```
   rm -rf /opt/pac/gui/work/Catalina/*
   ```
7. Delete user preferences files.

   This is required because Platform Application Center 8.0.1 user preferences file format is not compatible with the previous version's format. If you do not delete these files, users will not be able to see any lists. Note, however, that you will lose each user's preferences as a result.

   ```
   rm -fr $GUI_WORKDIR/listpreference_*.conf
   ```
8. Run RPM to install Platform Application Center 8.0.1.

For example:

```
rpm -Uvh pcc-appcenter-8.0.1-167869.x86_64.rpm
```

Use `--prefix` to install in a directory other than the default `/opt/pac`.

9. Check your configuration files.

   The update does not overwrite your existing configuration files. Compare your configuration files with the new configuration files in the `/opt/pac/gui/conf` directory. The new files have the extension rpmnew. For example, `pmc.conf.rpmnew`. To use any new features, you will need to copy over the new parameters to your configuration files.

10. Update your database schema.

    a) Untar the schema package.

    ```
    tar -xvf pcc-appcenter-8.0.1-dbschema.tar
    ```

    b) Apply changes to your database schema.

    - If you have a MySQL database, run the script in `DBschema/MySQL/Patch/`:

      Connect to your MySQL database with your existing user name and password.

      For example:

      ```
      mysql >source DBSchema/MySQL/Patch/pac80_to_801.sql;
      ```

    - If you have an Oracle database, run the script in `/DBschema/Oracle/Patch/`:

      ```
      sqlplus user_name/password@connect_string @DBSchema/Oracle/Patch/
      pac80_to_801.sql
      ```

      where

      - *user_name* is the user name on the database.
      - *password* is the password for this user name on the database.
      - *connect_string* is the named SQLNet connection for this database.

11. Start Platform Application Center services.

    ```
    pmcadmin start
    ```

    ```
    perfadmin start all
    ```

12. Check services have started.

    ```
    pmcadmin list
    ```

    ```
    perfadmin list
    ```

    You should be able to see the `WEBGUI`, `jobdt`, `plc`, and `purger` services started.

New in creating custom pages

Call an action from another action with <pac:call>

If an action was defined, it can be used in another action with the tag <pac:call>. For example:

```
<pac:action id="1" hidden="true">
        <![CDATA[
            ls/tmp
        ]]>
</pac:action>
<pac:action id="2" execute="true" result="text/plain">
        <![CDATA[
            echo  "action 2"
        ]]>
        <pac:call id="1"/>
</pac:action>
```

Redirect to the Flow Details page with <pac:jobflow-submit>

This tag redirects to the Flow Details page. Use it after you triggered a flow to display results. Only available when Platform Process Manager is installed.

<pac:jobflow-submit> has the following attributes:

Name	Description	Required
id	ID of the flow. Use the variable ${FLOWID}.	True
redirect	URL of the details page. Use the variable $JOBFLOW_DETAIL_URL?insId=${FLOWID} for the URL.	True

Example:

```
<pac:service id="test">
 ….
 <pac:action label="Submit" id="submit" result="text/xml">
        <![CDATA[
            echo "<pac:jobflow-submit id=\"${FLOWID}\"
            redirect='$JOBFLOW_DETAIL_URL?insId=${FLOWID}'/>"
        ]]>
      </pac:action>
</pac:service>
```

Redirect to the Job Details page with <pac:job-submit>

This tag redirects to the Job Details page. Use it after you submitted a job to display results.

<pac:job-submit> has the following attributes:

Name	Description	Required
id	ID of the job. Use the variable ${JOBID}.	True
redirect	URL of the details page. Use the variable $JOB_DETAIL_URL?jobId=${JOBID} for the URL.	True

Example:

```
<pac:service id="test">
....
<pac:action label="Submit" id="submit" result="text/xml">
        <![CDATA[
            echo "<pac:job-submit id=\"${JOBID}\"
            redirect='$JOB_DETAIL_URL?jobId=${JOBID}'/>"
        ] ]>
     </pac:action>
</pac:service>
```

Redirect to the Job Group Details page with <pac:jobgroup-submit>

This tag redirects to the Job Group Details page. Use it to display results after you submitted jobs associated with a job group.

`<pac:jobgroup-submit>` has the following attributes:

Name	Description	Required
id	ID of the job. Use the variable ${JOBID}.	True
redirect	URL of the details page. Use the variable $JOBGROUP_DETAIL_URL?groupName=${groupName} for the URL.	True

Example:

```
<pac:service id="JOBGROUP_CUSTOM_Sample_Service">
....
<pac:action label="Submit" id="submit_job" result="text/xml">
<![CDATA[
    JOBGROUPNAME="/test"
    JOB_COMMAND=`echo "$COMMAND" | awk -F":" '{ print $1}'`
    if [ "x$JOB_COMMAND"="x" ] ; then
            echo "Job command was not specified " 1>&2
            exit 1
    fi
    LSF_OPT=""
    if [ "x$JOB_NAME" != "x" ]; then
            LSF_OPT="-J \"$JOBGROUPNAME\""
    fi
    if [ "x$INP_INPUT_FILE" != "x" ]; then
            INPUT_FILE="\"${INP_INPUT_FILE}\""
            LSF_OPT="$LSF_OPT -i $INPUT_FILE"
    fi
    JOBID=""
    for n in {1,2,3,4,5}; do
            JOB_RESULT=`/bin/sh -c "bsub -g ${JOBGROUPNAME} ${LSF_OPT} ${JOB_COMMAND}
2>&1"`
            ID=`echo ${JOB_RESULT} | grep "<[0-9]*>" | sed 's/>.*$//' | sed 's/^.*<//'
`
            #get the JOB ID, if it's "", then return error message
            if [ "${ID}"="" ]; then
                    echo "$JOB_RESULT" 1>&2
                    exit 1
            fi
            #remember all sub job id into variable JOBID
            if [ "${JOBID}"="" ]; then
                    JOBID="$ID"
            else
                    JOBID="$JOBID,$ID"
            fi
    done
 echo "<pac:jobgroup-submit id=\"${JOBID}\" groupName=\"${GROUPNAME}\"
redirect='$JOBGROUP_DETAIL_URL?groupName=${GROUPNAME}'/>"
```

```
] ]>
    </pac:action>
</pac:service>
```

Redirect to the job directory with <pac:go-to-spooler>

This tag redirects to the job directory page. Use it during job submission.

Example:

```
<pac:spooler ttl="1d">
    <pac:service id="test">
        <pac:action execute="true">
        <![CDATA[
            echo "this is test log" > test.log
        ] ]>
        </pac:action>
        <pac:action label="Submit" id="submit" result="text/xml">
        <![CDATA[
            echo "<pac:go-to-spooler/>"
         ] ]>
        </pac:action>
    </pac:service>
</pac:spooler>
```

New built-in environment variables

- JOB_FLOW_DETAIL_URL: URL for the Flow Details page, to use after a flow is triggered.

  Example:

  ```
  echo "<pac:jobflow-submit id=\"${FLOWID}\" redirect='$JOBFLOW_DETAIL_URL?insId=$
  {FLOWID}'/>"
  ```

- JOB_DETAIL_URL: URL for the Job Details page, to use after a job is submitted.

  Example:

  ```
  echo "<pac:job-submit id=\"${JOBID}\" redirect='$JOB_DETAIL_URL?jobId=${JOBID}'/>"
  ```

- JOBGROUP_DETAIL_URL: URL for the Job Group Details page, to use after a job is submitted to a job group.

  Example:

  ```
  echo "<pac:jobgroup-submit id=\"${JOBID}\" groupName=\"${GROUPNAME}\"
  redirect='$JOBGROUP_DETAIL_URL?groupName=${GROUPNAME}'/>"
  ```

Platform Application Center log files and log levels

Use the Platform Application Center log file to troubleshoot errors that occur in the graphical user interface. Errors from add-ons such as Platform Process Manager, Platform Analytics, and Platform License Scheduler are also written to the log file.

Log file location

- `/pac/gui/logs/catalina.out`

Logging configuration file

Properties file location

The location of the logging configuration properties file is:

- `$GUI_CONFDIR/log4j.properties`

File format customization

The format of the log-file entries can be changed. For more details, see the log4cxx documentation:
`http://logging.apache.org/log4cxx/manual/classlog4cxx_1_1PatternLayout.html`

Log levels

The default log level is INFO.

Level	Description
DEBUG	Logs all debug-level messages.
	At this log level, all DEBUG, INFO, WARN, ERROR, and FATAL messages are logged.
INFO	Logs all informational messages about events that occurred.
	For example, Platform Application Center successfully started.
	At this log level, all INFO, WARN, ERROR, and FATAL messages are logged.
WARN	Logs only those messages that are potentially harmful. The system can tolerate this error. Features and performance are not affected.
	At this log level, all WARN, ERROR, and FATAL messages are logged.
ERROR	Logs only those messages that indicate error conditions . Includes system errors and user errors. These error conditions might allow the application to continue running. Features and performance may be partially affected.
	For example, cannot create a process to execute a job, failed to create directories because of disk quota, etc.
	At this log level, all ERROR and FATAL messages are logged.

Level	Description
FATAL	Logs only those messages in which the system is unusable. The whole system cannot work normally. For example, the database connection failed.
	At this log level, all FATAL messages are logged.

Submitting and controlling jobs in job groups

About job groups

In LSF, a collection of jobs can be organized into job groups for easy management. Users can submit, view, and control jobs according to their groups rather than looking at individual jobs.

You can submit, view and control jobs in job groups in Platform Application Center.

Job groups have the same access control and permissions as other LSF jobs.

For job data purging, when the system automatically purges job data, it checks if the job belongs to a job group. If some jobs in the job group do not have the status Done or Exited, the system does not purge any job data. When all jobs in the group have the status Done or Exited, data purging occurs in the same way as for other job data.

When you create your own custom pages, you use `<jobgroup-submit>` to display the Job Group Details page.

Submitting a job to a job group

In order for users to submit a job to a job group through Platform Application Center, you will need to change your application templates.

- To allow users to specify a job group at job submission:

  In your application template, create a text field and use the `bsub -g` option in your submission script so users can specify a new job group for a job.
- To enable users to select from an existing job group list when submitting a job:

  In your application template, add a new dropdown list field with type Job Group.

Monitoring jobs in a job group

- Job groups are listed in Jobs > By State. Use the Type column or Options > Filters to view your job groups.

  You can see the status of all jobs in a job group by selecting the job group. When you select a job group, you can see all the jobs in that group in the Jobs tab that is displayed. You can also view all data belonging to jobs in that job group in the Data tab in the Jobs page.
- Child job groups are displayed in their own entry. You can view child job groups through the Name column in the Jobs page.
- A job group's state reflects the state of jobs in the group. For example, if all jobs in the group are running, the job group's state is Running. However, if some jobs in the group are running and some have exited, you will see a striped state with both colors. Hovering with your mouse on the state color of the selected job group displays how many jobs are in each state.

Post and retrieve a job's external status

Use Web Services to retrieve the external status for a job

You can use web services to retrieve the external status for a job.

The output of `pacclient.py job` now has a new column: BSTATUS. This column displays the external status for a job.

You can also view this status in the Jobs page, with the Ext Status column. You will need to select the column for display in Options > Preferences

You use `bpost -d` to provide external status information for a job, and you use `bstatus` to retrieve that information, the Web Services Python API, or view that information through Platform Application Center. You can also use Web Services to retrieve that information.

Example:

```
bpost -d "Calculating Graph" 4006
bstatus 4006
JOBID FROM        UPDATE_TIME   STATUS
4006       userA Jun 21 17:27   Calculating Graph

C:\webservice\PAC8>pacclient.py job 4006
JOBID    STATUS    BSTATUS           JOB_NAME      COMMAND
 4006    RUNNING   Calculating Chart sleep 54321   sleep 54321
```

Python API Reference

The following functions in the Python API return the external status of a job:

- `getJobInfo(jobId)`
- `getJobForStatus(jobStatus)`
- `getJobForName(jobName)`

The output on success of these functions now includes `<extstatus>`.

status,message = getJobInfo (jobId)

Description

The `getJobInfo` API implements the get job web service. It shows job information for a job with the specified job ID.

Input

jobId

> Non-empty string
>
> Matches the LSF job ID

Output

On success

> (status="ok") "<Job> <id>%s</id><name>%s</name><status>%s</status>**<extStatus>%s</extStatus>**<cmd>%s</cmd><Job>"

On failure

> (status="error") error messages returned from the web service

status,message = getJobForStatus(jobStatus)

Description

The getJobForStatus API implements the get jobs for status web service. It shows job information for jobs having the specified job status..

Input

jobStatus

> One of the following:
>
> - Running
> - Pending
> - Suspended
> - Exited
> - Done

Output

On success

> (status="ok")
>
> "<Jobs><Job> <id>%s</id> <name>%s</name><status>%s</status>**<extStatus>%s</extStatus>**<cmd>%s</cmd><Job><Job> <id>%s</id> <name>%s</name><status>%s</status>**<extStatus>%s</extStatus>**<cmd>%s</cmd><Job>…</Jobs>"

On failure

> (status="error") error messages returned from the web service

status,message = getJobForName(jobName)

Description

The getJobForName API implements the get jobs for name web service. It shows job information for jobs matching the specified job name pattern.

Input

jobName

> Job name string pattern
>
> "*" is supported in the job name string to match multiple jobs

Output

On success

> (status="ok")

"<Jobs><Job> <id>%s</id> <name>%s</name><status>%s</status>**<extStatus>%s</extStatus>**<cmd>%s</cmd><Job><Job> <id>%s</id> <name>%s</name><status>%s</status>**<extStatus>%s</extStatus>**<cmd>%s</cmd><Job>…</Jobs>"

On failure

(status="error") error messages returned from the web service

For example: "Error to get Job info for job name: fluent*"