
Release Notes for Platform Process Manager

Platform Process Manager
Version 8.0.2
November 2011



Copyright

© 1994-2011 Platform Computing Corporation.

Although the information in this document has been carefully reviewed, Platform Computing Corporation (“Platform”) does not warrant it to be free of errors or omissions. Platform reserves the right to make corrections, updates, revisions or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY PLATFORM, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED “AS IS” AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL PLATFORM COMPUTING BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

We’d like to hear from you

You can help us make this document better by telling us what you think of the content, organization, and usefulness of the information. If you find an error, or just want to make a suggestion for improving this document, please address your comments to doc@platform.com.

Your comments should pertain only to Platform documentation. For product support, contact support@platform.com.

Document redistribution and translation

This document is protected by copyright and you may not redistribute or translate it into another language, in part or in whole.

Internal redistribution

You may only redistribute this document internally within your organization (for example, on an intranet) provided that you continue to check the Platform Web site for updates and update your version of the documentation. You may not make it available to your organization over the Internet.

Trademarks

LSF is a registered trademark of Platform Computing Corporation in the United States and in other jurisdictions.

ACCELERATING INTELLIGENCE, PLATFORM COMPUTING, PLATFORM SYMPHONY, PLATFORM JOB SCHEDULER, PLATFORM ISF, PLATFORM ENTERPRISE GRID ORCHESTRATOR, PLATFORM EGO, and the PLATFORM and PLATFORM LSF logos are trademarks of Platform Computing Corporation in the United States and in other jurisdictions.

UNIX is a registered trademark of The Open Group in the United States and in other jurisdictions.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

Third-party license agreements

<http://www.platform.com/Company/third.part.license.htm>

Contents

| | |
|--|----|
| Platform Process Manager Files | 5 |
| General new features | 6 |
| Enhancements to local jobs on Linux and UNIX | 6 |
| New built-in user variable JS_FLOW_FULL_NAME | 6 |
| Use a custom mail program to send email | 7 |
| Restrict who can see the flow chart view | 7 |
| New features in Flow Manager | 8 |
| New features in Flow Editor | 10 |
| Flow-related new features available only in Platform Application Center | 14 |
| Jobs and Flows can now be monitored in the same window | 14 |
| Completion attributes now visible for subflows and flow arrays in Flow Chart tab | 14 |
| Reorganization of pages for flow definitions | 14 |
| Installation Options | 16 |
| Known Issues | 17 |

Platform Process Manager Files

| Documentation | File Name |
|--|------------------------|
| <i>System Requirements</i> | sys_requirements.html |
| <i>Installing Platform Process Manager (UNIX)</i> | pm_install_unix.pdf |
| <i>Installing Platform Process Manager (Windows)</i> | pm_install_windows.pdf |
| <i>Administering Platform Process Manager</i> | pm_admin.pdf |
| <i>Using Platform Process Manager</i> | pm_using.pdf |

General new features

Enhancements to local jobs on Linux and UNIX

This feature is available in:

- Platform Process Manager

Description:

These enhancements to local jobs only apply to Linux and UNIX.

A local job is a job that will execute immediately on the Platform Process Manager host without going through LSF. A local job is usually a short and small job.

Enhancements that have been made:

- Local jobs are now non-blocking. This means that multiple local jobs can run at the same time.
- You can now kill a local job. If a local job is killed outside of Process Manager, Process Manager can identify the local job's exit status and resource usage.
- Local jobs are now suspended and resumed when you suspend or resume the flow that contains them.
- In the job's runtime attributes, you can now view the exit status and CPU usage of a local job after the job completes. The process ID identifies the local job and you can view CPU usage for the job. You can also view the process ID of the job and CPU usage information with `jobs -l flow_id` and `jobs -C job`.
- To avoid overloading the Platform Process Manager host with too many local jobs, there is a new parameter `JS_LOCAL_JOBS_LIMIT` in `js.conf` to control the maximum number of local jobs that can simultaneously run on the Process Manager host.
- By default, local jobs now have no timeout. The default value of `JS_LOCAL_EXECUTION_TIMEOUT` in `js.conf` has been changed to unlimited.
- The parameter `JS_LOCAL_EXECUTION_THREADS` in `js.conf` is now obsolete. Its value is now fixed at 1 and cannot be changed, as local jobs are now non-blocking.
- Should `jfd` terminate abnormally, when it restarts it can recover running and finished local jobs and determine their status and resource usage.
- A new binary is installed in `JS_SERVERDIR: eem.local`. It is started by `jfd` and handles job submission, control, and status checking for local jobs and reports back to `jfd`.
- Two additional port numbers are now used by `jfd` and `eem.local`: `JS_PORT + 1` and `JS_PORT + 2`.

New built-in user variable

JS_FLOW_FULL_NAME

This feature is available in: Platform Process Manager and Platform Application Center.

Description:

You use the built-in user variable `JS_FLOW_FULL_NAME` when you need to use the long version of a subflow name.

For example:

- For a subflow named 11:usr1:F1:SF1:SSF1, this variable is set to 11:usr1:F1:SF1:SSF1.
- For a main flow named 11:usr1:F1, this variable is set to 11:usr1:F1.

Use a custom mail program to send email

This feature is available in:

- Platform Process Manager: Set `JS_MAILPROG` in `js.conf` to your custom mail program. After setting your custom mail program, you will need to restart `jfd` with the commands `jadmin start` and `jadmin stop` to make changes take effect.

Description:

By default, Process Manager sends email through `/usr/lib/sendmail` on UNIX or `lsmail.exe` on Windows.

You can now specify a custom mail program to send emails. Your custom mail program can be a shell script, a binary executable, or, a `.bat` file on Windows. Your custom mail program must follow the same protocol as `sendmail`.

Restrict who can see the flow chart view

This feature is available in:

- Platform Process Manager: You set the parameter `JS_LIMIT_FLOW_CHART_VIEW` in `js.conf` and affects display of the flow chart and associated actions in Flow Manager and Platform Application Center.
- Platform Application Center: Flow Chart view is restricted along with associated actions based on the parameter set in Platform Process Manager.

Description:

There is a new parameter in `js.conf`, `JS_LIMIT_FLOW_CHART_VIEW`. This parameter allows you to restrict viewing the chart view of a flow and flow definition to only the Process Manager administrator and users who are both the flow definition owner and flow owner.

When this parameter is set to `false`, users who can view a flow or flow definition, can see everything about the flow: flow chart, general information, subflows and jobs, flow data, and flow history. These users can also perform job and subflow-specific actions.

When this parameter is set to `true`, there are restrictions on which users can see the flow chart of a flow and flow definition and associated actions the user can take on components of the flow.

New features in Flow Manager

Hold and release for jobs

This feature is available in:

- **Flow Manager:** You can hold and release jobs through Flow Manager By State tab, display a flow, and select the job in the Waiting state, right-click and choose Hold, or the new options in the `jjob` command, `-p` for hold, and `-g` for release.
- **Platform Application Center:** Go to Jobs > Jobs > By State > Running, select the flow, select the Flow Chart tab, select the job in the Waiting state, right-click, and choose Hold.

Description:

In some cases, you may want to stop a flow at a specific point so that you can fix problems. You can do this by putting a job in the Waiting state in the flow on hold.

Only the branch of the flow that contains the job that is On Hold pauses. Other branches of the flow continue to run.

You can put on hold LSF jobs, job submission scripts, local jobs and job arrays.

Allow users to trigger other users' flows

This feature is available in:

- **Platform Process Manager:** Set `JS_CHANGE_FLOW_OWNER` in `js.conf`. There is also now one more tab in Flow Manager, the By Definition tab. This tab displays flow definitions organized by the user who submitted them. In addition, what is displayed in the tree view has been enhanced for all tabs to indicate the flow owner and flow submitter.
- **Platform Application Center:** The parameter setting in Process Manager also controls who can trigger other users' flows in Platform Application Center. Select Jobs > Flow Definitions > By User to trigger flows.

Description:

By default, only Process Manager administrators and Process Manager control administrators can trigger flows created by other users.

This feature only applies to flow definitions that have the status Published.

With the new parameter `JS_CHANGE_FLOW_OWNER=true` in `js.conf`, non-administrator users can trigger other users' flows. In this way, one user can submit flow definitions, and another user can trigger the flow from the flow definition, own the flow, and control it. The user who submitted the flow definition is the owner of the flow definition, the user who triggered the flow is the owner of the flow.

Rerun a flow while a job is still running

This feature is available in:

- **Platform Process Manager:** In Flow Manager, the Rerun Now and Rerun with variables menu items have been replaced with Rerun, and a window is displayed in which you can choose what to rerun in the flow.

- Platform Application Center: Select Jobs > Jobs > By State, select the flow, and click the Rerun button to display a window in which you can choose what to rerun in the flow.

Description:

In previous versions, you could only rerun flows that were in an Exited state. You can now rerun flows when the flow state is Running, Exited, or Done.

This is useful for flows that have several branches. When one branch fails, you can rerun the branch without waiting for other branches of the flow to complete.

You can:

- Set or unset starting points when there are still jobs running in the flow.
- Choose whether to rerun the flow from:
 - Starting points and exited jobs. The flow will rerun from any starting points, exited jobs, and, from the item following any manually completed jobs provided dependencies are met.
 - Starting points only. The flow will rerun only from starting points.

Note that you can only rerun a running flow if the part of the flow to be rerun does not overlap with items that are currently running.

Exit codes for manual jobs

This feature is available in:

- Platform Process Manager: In Flow Manager, you can now specify exit codes when completing a manual job, or by using the new option in the `job complete` command, `-e exit_code`. Manual jobs can now fail. In Flow Editor, you can now specify in the manual Job Event Definition the dependencies Fails, Ends with any exit code, and Ends with exit code....
- Platform Application Center: You can complete a manual job and specify an exit code through Jobs > Jobs > By State > Pending User Input, select the manual job, click the Complete Manual Job button.

New features in Flow Editor

New Other Options field for additional LSF job submission options

This feature is available in:

- Flow Editor: Job Definition or Job Array Definition dialog, Advanced tab, Other Options field.
- Platform Application Center: In the Job Definition or Job Array Definition dialog, Advanced tab, Other Options field. To access the Job Definition, in the Jobs tab, select Jobs, select a Flow, select the Flow Chart tab, right-click a job to display the Job Definition.

Description:

This allows you to use options that are not available from the job definition dialog. The options you specify are added to the `bsub` command when you submit the job or job array.

For example:

```
-w "done("#{JS_FLOW_FULL_NAME}:JobArray1")"
```

You can also specify user variables in the Other Options field.

Configure custom exit codes for successful jobs

This feature is available in:

- Platform Process Manager: In Flow Editor, open the Job Definition dialog, Job Script Definition dialog, Manual Job Definition dialog, or Local Job Definition dialog, and configure the new field Non-zero success exit codes.
- Platform Application Center: You can view settings for the field Non-zero success exit codes in the Job Definition, but you cannot change them.

Description:

By default, for a job to complete successfully, the exit code must be 0. Any other exit code indicates the job failed.

In some cases, however, you may want to use exit codes to pass information to subsequent work items and may want to use numbers other than 0 to indicate success.

You can now do so by specifying a space-separated list of exit codes in the Job Definition dialog, Job Script Definition dialog, Manual Job Definition dialog, or Local Job Definition dialog, with the new Non-zero success exit codes field.

Configure how to calculate flow exit codes

This feature is available in:

- Platform Process Manager: In Flow Editor, select Action > Specify Flow Completion Attributes, new section Determine the flow exit code from
- Platform Application Center: You can view Flow Completion Attributes but you cannot change them.

Description:

By default, a Done flow or subflow has an exit code of 0, since the default way that Process Manager determines the flow exit code is through the sum of all exit codes of all work items in the flow.

However, it is possible to specify custom success exit codes for LSF jobs, job scripts, local jobs, and manual jobs. For this case, you can configure the flow to inherit the exit code of the last item that was successfully completed or that failed in the Flow Completion Attributes dialog.

New dependencies

This feature is available in:

- Platform Process Manager: In Flow Editor, new dependencies have been added for subflows, flow arrays, and jobs.

Flow Event Definition, for subflows:

- The flow completes successfully with exit code...
- The flow fails with exit code...
- The flow fails

Flow Array Event Definition:

- Any flow fails

Job Event Definition:

- Fails
- Is Submitted

Job Array Event Definition:

- Any job fails
- Platform Application Center: You can view dependency settings but you cannot change them.

User variables in more fields when defining jobs and job arrays

This feature is available in:

- Platform Process Manager: In Flow Editor, Job Definition and Job Array Definition dialogs.
- Platform Application Center: User variables are displayed, but cannot be specified.

Description:

You can now use user variables in more fields in the Job Definition and Job Array Definition dialogs. When you select a field and hover, the help that displays indicates whether you can use a user variable or not in the field.

User variables for job parameters are resolved at runtime, just before the job is submitted.

The following fields now support user variables:

| Tab | Field |
|----------------|---|
| Processing tab | Number of Processors for Parallel Jobs, Minimum |

| Tab | Field |
|------------|---|
| | Number of Processors for Parallel Jobs, Maximum |
| | Before Execution, Run command |
| | User Group, Associate job with user group |
| Limits tab | All fields under Job Limits |
| | Host Limits, Maximum run time |
| | Host Limits, Maximum CPU time |

Submit a dependent job after selected jobs start running or are submitted

This feature is available in:

- Platform Process Manager

Description:

- In Flow Editor, Advanced tab, Pre-submit section, you can now select jobs upon the current job depends. This now applies not only to jobs and job scripts, but also to job arrays, job array scripts, and template jobs.
- You can now specify either Starts or Submitted as the dependency. In this way, you can identify that the current job is to be submitted right after the selected jobs have started to run in LSF, or that the current job is to be submitted right after the selected jobs have been submitted to LSF.
- Create proxy events for jobs with the new Starts or Is Submitted events
- Create proxy events for job arrays with the new Number of jobs started is..., and The job array is submitted events.

Static and dynamic flow arrays can now run sequentially

This feature is available in:

- Platform Process Manager: In Flow Editor, Flow Array Attributes dialog.

Description:

In Flow Editor, there is now an option in the Flow Array Attributes to run in parallel or sequentially. As a result, you now have the choice of running static or dynamic flow array elements in parallel, or sequentially. In previous versions, flow arrays always ran in parallel.

Determining success or failure based on specific exit codes in the dependency condition

This feature is available in:

- Platform Process Manager: In Flow Editor, Job Event Definition, Proxy Event Definition, and Exception Handler Definition with the events Ends with exit code equal to and Ends with Exit code not equal to.

Description:

You can now define dependencies to take action if any of the specified exit codes are encountered.

You can specify a list of exit codes in:

- Dependencies between jobs, job scripts, template jobs, local jobs, and manual jobs.
- Proxy event definitions for a proxy job, proxy template job, proxy job script, and proxy local job. For proxy dependencies, you can also use `j sub - p` and specify a list of exit codes.
- Exception Handler Definition for a job, job script, template job.

Command to run field can now display multiple lines

This feature is available in:

- Platform Process Manager: Flow Editor, in the definition of a job, job array, or local job.
- Platform Application Center: You can enter the command to run in Jobs > Submission Forms > Flow Forms. You can view a command that spans multiple lines in Jobs > By State, by selecting the state, selecting a flow, selecting the Flow Chart tab, right-clicking and selecting Open Definition for a job, job array, or local job.

Flow-related new features available only in Platform Application Center

Jobs and Flows can now be monitored in the same window

This feature is available in: Platform Application Center.

Description:

Jobs and flows are now in the same window, accessible through Jobs > Jobs > By State. There is now a Type column by which you can sort.

Possible types are:

- Job
- Flow
- Array

Completion attributes now visible for subflows and flow arrays in Flow Chart tab

This feature is available in: Platform Application Center.

Description:

You can now view completion attributes for static and dynamic subflows, and flow attributes and completion attributes for static and dynamic flow arrays.

For subflows, select Jobs > Submission Forms > Flow Forms, select a flow, select the Flow Chart tab, select a subflow, right-click and choose Completion Attributes.

For flow arrays, select Jobs > Submission Forms > Flow Forms, select a flow, select the Flow Chart tab, select the flow array, right-click and select Expand. When the new page is displayed, right-click on the page, and select the Attributes or Completion Attributes menu items.

Reorganization of pages for flow definitions

This feature is available in:

- Platform Application Center: Pages related to flow definitions have been reorganized.

Description:

- Resources > Submission Templates > Flow Definitions: view flow definitions as a list or graphically and perform actions on the flow definitions: Hold, Release, Remove, Publish, Unpublish.

- Settings > System Services > Flow Manager Service: View the Process Manager server name, port, and statistics about the number of flows and flow definitions in each state, and set global variables for all flows.
- Jobs > Submission Forms > Flow Forms by User: Trigger a flow from a flow definition. Non-administrator users can see their own submitted flow definitions and all published flow definitions. Process Manager administrators and control administrators can see all submitted flow definitions and flows.
- Jobs > By State > Pending User Input: View and complete manual jobs.
- Jobs > Job Alerts: View open alarms in the system.

Installation Options

Default installation

For a default installation on either Windows or UNIX, follow the applicable installation instructions in the following guides:

| Book | File |
|--|------------------------|
| <i>Installing Platform Process Manager (Windows)</i> | pm_install_windows.pdf |
| <i>Installing Platform Process Manager (UNIX)</i> | pm_install_unix.pdf |

Control the JFD

- To start JFD:
 - UNIX: `jadmin start`
 - Windows:
 1. From the Windows Control Panel, select Administrative Tools > Services.
 2. Right-click Process Manager and select Start.
- To stop JFD, run `jadmin stop`. Do not manually kill the `jfd` process.

Known Issues

Platform Process Manager 8.1 has the following known issues:

- On Linux, jfd requires the `libstdc++.so.5` library. If this is not found on your system, install `libstdc++` or the `compat-libstdc++` RPM containing the `libstdc++.so.5` library that is appropriate to your system.
- `bhist` and `bacct` do not work on Windows hosts running in a UNIX cluster.

This occurs because Windows hosts cannot read the shared directory on UNIX. To work around this issue, run `bhist` or `bacct` from a UNIX host in the cluster.

- If you installed YouDao (translation software), dialog windows in the Calendar Editor may disappear when you double-click or drag the window, or drag a scroll bar in the window. This problem occurs with every item in the Calendar Editor except Calendar > New Calendar > Combine calendars.

For example, when in the Owners > Sys directory tree, if you double-click Daily, the Daily@Sys window displays in the main window. If you then double-click or drag this window, it will disappear. If this window has a scroll bar, the window will also disappear if you drag the scroll bar.

This is a problem with the YouDao software. To work around this issue, shut down YouDao.

- When a user variable is used in a job definition's Name field, the value of the variable must be set when the flow is triggered. Otherwise the flow will fail to be triggered. For example, in Flow Manager, you may choose "Trigger" => "With Variables", or on command line use `jtrigger -v variable_list ...`.
- Applies to Linux hosts when `JS_TIME_ZONE` is set to "client", the default. During Daylight Savings Time (in the March-November time period), some time events may trigger an hour early. This occurs when Java gets the wrong timezone information from the client machine. Solution: On the (Linux) client, edit the file `/etc/sysconfig/clock` and replace `TIMEZONE=zone` with `ZONE=zone`. For example, `ZONE="US/Eastern"`.
- On Linux Redhat 7.0, due to a known java problem, scripts that run Platform Process Manager commands and then monitor for exit code will never see an exit code 1 when it occurs—they see an exit code of 0.
- On HP, the failover feature does not work with a permanent license. Use a demo license.
- A Misschedule exception is not propagated through two consecutive link events in a flow.
- When using the Flow Editor or Flow Manager, and viewing or changing a job or job array definition, the entire job or job array definition dialog cannot be viewed if you are using a screen resolution lower than 1024 x 768. We recommend you set your resolution to 1024 x 768 or higher when using the Platform Process Manager graphical user interface.
- When launching the graphical interfaces (Flow Editor, Calendar Editor and Flow Manager) on some UNIX systems, the user receives font not found warnings. The Java Virtual Machine user interface manager automatically determines which fonts are used on each machine. If the fonts are not available on the machine, the warning is issued. These warning messages do not affect the operation of the client.
- Platform Process Manager does not validate the end date of a calendar expression. Consequently, if you define an end date prior to the start date, you receive no error message, but the calendar cannot be used in a time event to successfully trigger a flow or job.
- If you kill a job from the Flow Manager, the job state does not automatically refresh to reflect the change—you need to refresh manually, unless real-time status updates are being used.
- In the Flow Editor, on the Job Definition tabs, the Reset button only resets data on the current tab—it does not affect the hidden tabs. Values specified in the following fields do not reset at all when you click the Reset button:

- Submit to queue/partition(s)—General tab
- User name—General tab
- Host requirements fields—Processing tab
- Some days of the week are not completely shown in the Calendar Editor for some platforms:
 - On Solaris: Mon, Wed, Thu are shown as M..., W..., T...
 - On HP: Mon, Wed are shown as M..., W...
- Some historical information is not logged correctly in the following circumstances:
 - If a job or other activity has a rerun exception handler that is triggered, no information about the exit code of the job is logged.
 - When user variables are used for file transfer names, the resolved name of the file is not logged.
- When using the `j sub - r` command to submit and replace a flow definition, if the `j sub` command fails, the replacement does not occur, but the replaced flow definition is already deleted and cannot be restored.
- In the Flow Manager, it is possible for the user to have two different flows selected in the tree view and the right-hand pane. If the user selects an option in the Action menu, such as Kill, it is the flow selected in the tree view that is killed, not the one in the right-hand pane. This may not be the behavior the user expects.
- Platform Process Manager does not support duplicate event logging in LSF. The `lsb.events` file must be in a shared location where the Platform Process Manager server can locate job information.
- If a job using a rerun exception handler has a rerun delay specified, and the Platform Process Manager restarts during the delay period (prior to rerunning the job), the job is never rerun.
- If a user tries to open the job dependency definition on the arrow coming out of the job with the variable in the job name field, the user sees an empty message dialog, prior to seeing the dependency definition. The user can click OK, to see the definition. However, the job name field is blank.
- If a job command (command itself, not parameters) contains space and if the job is submitted to a queue with a job starter, the command doesn't work.