# Release Notes for Platform Process Manager

Platform Process Manager
Version 8.0
November 2010

Platform
Computing

### Copyright

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

### 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

# Process Manager Files

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

# New features

## New method of setting user variables

- You can now use external files to set user variables for the following work items:
    - Jobs and job arrays
    - Job submission scripts and job submission script arrays
    - Local jobs

  These external files are specified by the JS_FLOW_VARIABLE_FILE environment variable for flow variables, by the JS_GLOBAL_VARIABLE_FILE environment variable for global variables, and by the JS_PARENT_FLOW_VARIABLE_FILE environment variable for parent flow variables.

  This external file contains one variable-value pair on each line. The values in the external file must not contain semicolons (;) or control characters.

  Process Manager will not initially create these files — the jobs need to create these files. This approach does not require a job starter, and the job command is not required to use shell scripts. Any binary or script will work — they just need to write to the file.

  The original method of setting flow and global variables (JS_FLOW_VARIABLE_LIST and JS_GLOBAL_VARIABLE_LIST, respectively) still work, but are now deprecated and Process Manager uses this original method internally to obtain the variable-value pairs.
- When setting parent flow variables, you can now use the JS_FLOW_SHORT_NAME built-in variable when you need to use the shortened version of the flow name to avoid a potential name conflict issue when using JS_PARENT_FLOW_VARIABLE_FILE to set parent flow variables.

  For example, there are two dynamic subflows (`DSF1` and `DSF2`) in a main flow (`11:usr1:F1`), that both refer to the same target flow (`TF`). If the target flow sets a parent flow variable `myvar`, both dynamic subflows will overwrite each other's value of the `myvar` variable.

  To prevent this issue, for all subflows and flow arrays in a flow instance, use the JS_FLOW_SHORT_NAME variable to indicate the name of the subflow.

  For example,

    - For a subflow named `11:usr1:F1:SF1:SSF1`, this variable is set to `SSF1`.
    - For a main flow named `11:usr1:F1`, this variable is set to `F1`.
    - For a flow array element named `11:usr1:F1:FA(1)`, this variable is set to `FA`. Note that this does not include the array index. If you need to differentiate between array elements, you must use the `JS_FLOW_INDEX` built-in user variable.

## Using the variable evaluator for decision branch variables

- Using the flow editor, you can now add a new work item to a job flow called the Variable Evaluator (VE). This new work item allows jobs to depend on the evaluation of variable expressions instead of the traditional job exit status, time events, file events, and proxy events. This new work item contains no actual jobs to run, and its purpose is to serve as an intermediate step between jobs and the validation of variable decision branches.

In a typical work flow, the predecessors of a variable evaluator assign values to user variables. When all the variables are set, the variable evaluator will then evaluate all of its variable expression branches and determine which sucessors should start executing.

# Flow arrays

- Using the flow editor, you can now add flow arrays as part of a job flow. A flow array works in a similar fashion to a job array, but at a subflow level. A flow array follows the same convention as a subflow and requires a user to select a flow definition to form the array elements. When Process Manager instantiates the flow array, a specified number of flow elements start to run in parallel.

# Local jobs

- Using the flow editor, you can now add local jobs as part of a job flow. Local jobs will execute immediately on the Process Manager host without going through LSF.

  Since a local job is short, killing a local job is disabled in Process Manager to simplify the functionality of a local job.

# Dynamic subflows and flow arrays

- Using the flow editor, you can now add an existing flow definition as a dynamic subflow within a flow diagram.

  A dynamic subflow, also called a subflow by reference, refers to a target flow that has already been submitted to Process Manager. Only flows that have been submitted to Process Manager and published are eligible target flows for dynamic subflows. When you modify the definition of the target flow, all the mainflows that reference this target flow can obtain the latest version.
- Using the flow editor, you can now add an existing flow array definition as a dynamic flow array within a flow diagram.

  A dynamic flow array, also called a flow array by reference, refers to a target flow that has already been submitted to Process Manager. Only flows that have been submitted to Process Manager and published are eligible target flows for dynamic flow arrays. The target flow is essentially converted into a dynamic flow array in which each array element is equivalent to the target flow.
- To distinguish between the dynamic subflows or flow arrays and the original (non-dynamic) subflows or flow arrays, the original (non-dynamic) subflows are now referred to as static subflows, while the original (non-dynamic) flow arrays are now referred to as static flow arrays.

# Flow versioning

- You can now submit a flow with comments to provide a description of each flow version. This makes it easier to track different versions of the flow.
- Using the flow manager, you can now view the version history of the flow to see the different versions of the flow that are submitted. In the flow manager, right-click a flow definition and select View Version, then click View History to view the version history list.
- Using the flow manager, you can now set the default version of the flow. The default version of the flow is the version set to be effective at the current time. If you trigger this flow, Process Manager will instantiate the flow instance with the default version.

  In a dynamic subflow that is automatically updated, the currently-used version is the same as the default version. In a dynamic subflow that is manually updated, the currently-used version is the default version of the target flow at the main flow submission time, or the default version at the time that you last manually updated the dynamic subflow.

- You can now set the default version of the specified flow using the `jsetversion` command.
- The `jflows -l` command now shows the currently-used version of the specified flow (specifically, **jflows -l -f *flow_name***).
- The `jdefs -l` command now shows the default version and the latest version of the specified flow (specifically, **jdefs -l -f *flow_name***).
- You can now view the version history of a flow by using the `jdefs -v` command.

# Defining and viewing input variables in flows and jobs

- You can now specify user variables (in addition to environment variables) when specifying flow attributes to allow a flow to use these variables. Previously, you could only define environment variables, but now you can define these variables as input variables.
- You can now view the name and values of input and local variables when viewing the Runtime Attributes of a flow, or by using the `-l` option of the `jjob` command.
- Using the flow manager, you can now set user variables when triggering flows by using the Trigger > With Variables menu option.
- Input variables are now available to static and dynamic subflows, and flow arrays.
- Input variables for jobs are now referred to as environment variables.

# Viewing and setting flow variables

- Using the flow manager, you can now view and set flow variables for subflows and flow arrays in addition to main flows.
- In the flow manager, the Set Variables menu item is now Set Flow Variables to reflect the fact that this menu item is now available to subflows and flow arrays (in addition to flows). Navigate to this menu item by by right-clicking on a flow instance that has a status of Exit, Running, or Suspended.
- The `jsetvars` command now supports the `-s`, `-r`, and `-l` options for use with `-i` *flow_ID*. This means that you can now add, edit, remove, or view flow variables (in addition to global variables by using the `-g` option).
- The `jsetvars` command now allows you to specify the scope for the `-s`, `-r`, and `-l` options. This means that you can now specify a specific scope (such as for a specific subflow) when adding, editing, removing, or viewing flow variables.

# Setting additional starting points to rerun flows

- Using the flow manager, you can set specific work items in the flow from which to rerun the flow. This allows you to have more flexibility in correcting errors in a flow by rerunning jobs other than the last exited job in the flow.

  Exited jobs in the flow automatically become rerun starting points in addition to any rerun starting points that you have set.

# Displaying jobs that are pending in LSF

- Process Manager now displays a brown border around jobs that are pending in LSF in a flow, which means that the job is waiting in a queue for scheduling and dispatch. Previously, Process Manager treated these jobs as Running and displayed a green border around these jobs. Tooltips and job runtime attributes show these jobs as Pending in LSF, while `jflows -l` output shows these jobs as `Pending`.

Unless otherwise specified, any operation that applies to Running jobs also applies to jobs that are Pending in LSF.

# Flow working directory

- You can now specify the working directory for flows, subflows, and flow arrays (in addition to jobs and job arrays). All valid inner work items (subflows, jobs, and job arrays) in the flow will use this directory as the working directory unless you further specify a working directory for the inner work item. In this case, the working directory setting for the inner work item will override the setting for this flow.

  As for job and job array working directories, you can use user variables when specifying the flow working directory.

# Multiple Process Manager servers

- You can now run multiple Process Manager servers in a single Platform LSF cluster. This is useful, for example, if you have different Process Manager environments running in one cluster.

  To avoid conflicts and to ensure that each job is unique among the multiple Process Manager servers, you must ensure that each combination of user name and flow name is unique within the cluster.

# New version of JRE

- JRE has changed from version 1.4 to 1.6.

# Deprecated support for Solaris 7

- Process Manager no longer supports Solaris 7.

# Deprecated parameters and variables

## Environment variables

The following environment variables are deprecated in Process Manager 8.0.

| Old environment variable | Description | Replaced with |
|---|---|---|
| JS_FLOW_VARIABLE_LIST | The list of flow variable-value pairs. | The JS_FLOW_VARIABLE_FILE environment variable, which points to the text file containing the list of flow variable-value pairs. |
| JS_GLOBAL_VARIABLE_LIST | The list of global variable-value pairs. | The JS_GLOBAL_VARIABLE_FILE environment variable, which points to the text file containing the list of global variable-value pairs. |

# Installation Options

## Default installation

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

| Book | File |
|---|---|
| *Installing Platform Process Manager (Windows)* | `pm_install_windows.pdf` |
| *Installing Platform Process Manager (UNIX)* | `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

Process Manager 8.0 has the following known issues:

- 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”.
- The SGI IRIX 6.5.24 platform does not support the new Daylight Savings Time dates (that changed in 2007) when JS_TIME_ZONE is set to "client" (the default). Time-based scheduling is affected.
- On Linux Redhat 7.0, due to a known java problem, scripts that run 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 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.
- 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...
  - On SGI: Sun, Mon, Wed are shown as S..., 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 `jsub -r` command to submit and replace a flow definition, if the jsub 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.
- Process Manager does not support duplicate event logging in LSF. The `lsb.events` file must be in a shared location where the Process Manager server can locate job information.
- If a job using a rerun exception handler has a rerun delay specified, and the 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.
- in xlsadmin, some menu items under Manage may be disabled after operating shutdown or close actions against LSF daemons.
- In xlsadmin, lock/unlock LIM and start/stop service only has effect on the LIM that locates on the host where xlsadmin is started.
- in xlsadmin, some menu items under Manage may be disabled when LSF_EGO_DAEMON_CONTROL="Y" is set in lsf.conf