Release Notes

Platform Symphony™
Version 5.0
Release date: November 2009
Last modified: November 16, 2009

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

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 JOBSCHEDULER, 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

What’s New?

Resource Management

Selective Reclaim

This feature ensures the least impact to existing workload when other applications require their share of the resources. By default, Platform Symphony will reclaim resources as quickly as possible with minimum overhead. However, this does not take into account the importance of sessions and length of time a task has been executing on a particular resource. The Selective Reclaim feature enables the system to consider the relative importance of running work and to reclaim resources from the least important session(s) first. Once the least important session is identified, any recently started tasks associated with that session can be stopped.

Refer to *Configuration to enable selective reclaim* in the *Cluster and Application Management Guide*.

Configurable Resource Distribution Order: Planned Resource First for Multiple Resource Group

A new (configurable) resource distribution policy is added to Platform Symphony where the planned number of slots from each resource group are distributed to the consumer first, then the unused portion of slots for each resource group from other consumers are distributed to the consumers with demand, to the maximum limit allowed by the resource plan. This feature could potentially reduce the number of reshuffles (hosts reclaims) in the cluster.

Refer to *Using Static Share Quota to Minimize Reclaims Across Multiple Resource Groups* in the *Cluster and Application Management Guide*.

Exclusive Host at Resource Group Level

This feature allows the hosts in the resource group to be distributed as a whole rather than as slots to the workload manager (SSM) and allows SSM to distribute the slots to application sessions in the most optimal way. This feature could reduce slot fragmentation especially when used in conjunction with the Dynamic Slot feature.

Refer to *Exclusive allocation* in the *Cluster and Application Management Guide*.

Graceful shutdown of custom EGO Services

This feature enables Platform EGO services to specify custom script to gracefully shutdown service instances. A configurable timeout parameter instructs the systems when to kill target instances if they are still running after the timeout expires.

Refer to *About system services* in the *Cluster and Application Management Guide*.

Workload Management

Priority FCFS Scheduling

Priority FCFS Scheduling is a new scheduling policy that ensures high-priority sessions with time-critical tasks receive as many available resources as they can use in order to finish as soon as possible. Tasks from lower priority sessions may be preempted. If more than one session has the same priority, sessions created earlier would be considered high-priority.

Refer to *How resources are scheduled by the session manager* in the *Cluster and Application Management Guide*.

Task Preemption

This feature allows urgent work to be processed right away rather than waiting for previous workload to finish. You can configure a session to be preemptive so that when the session is under-allocated, it can preempt workload of other sessions instead of waiting for other sessions to voluntarily release slots. Platform Symphony takes back resources from lower priority sessions and gives them to higher priority sessions by terminating the currently running tasks of the lower priority sessions.

Refer to *Preemption* in the *Cluster and Application Management Guide*.

Session-based Resource Preference

This feature provides a means for the application developer to state a preference for service and host attributes that provide the best match for the workload in a session. In a typical cluster environment, all machines are not identical. Some machines may be slower and some machines may have less memory than others. If you configure a resource requirement at the application level, all sessions for that application have to use the same set of resources. However, it may not be the case that all sessions are the best match for the resources. Some sessions may benefit from special resources so that results can be computed in the most optimal way.

With the resource preference feature you can specify a preferential association between a session and a host that possesses the attributes best suited to process the workload. The association is based on the evaluation of a user-defined expression containing resource attributes capable of being monitored. The evaluation of this expression is carried out against the attributes of each host available to the session.

Refer to *Resource-aware scheduling for sessions* in the *Cluster and Application Management Guide.*

Multiple Resource Group Selection at Application level

Now you can specify multiple resource groups rather than "all" or just one resource group. This feature allows applications to be more selective about the resource groups that can be used.

Session-level Resource Group filter

In previous versions of Platform Symphony, applications can specify types of resources that should be used by all sessions in the application by listing the resource group in the application profile. In Platform Symphony 5.0, in addition to application level resource group filters, each session can specify a resource group filter that can further narrow down the choices to specific types of resources.

Refer to *About slot usage requirements and resource group filtering* in the *Cluster and Application Management Guide.*

Delay Slot Release

To maximize the utilization of resources, Platform Symphony, by default, releases resources for other applications to use as soon as there are no tasks pending for this particular application. This feature allows Platform Symphony to keep the resources and Services Instances running on those resources for a configurable time even if there are no pending tasks for the application.

Refer to *delaySlotRelease* in the *Reference Guide.*

Standby Service

To maximize the utilization of resources, Platform Symphony, by default, releases resources for other applications to use as soon as there are no tasks pending for this particular application. Each time the resources are released, service instances on these resources are terminated. When more tasks are received, new resources are allocated and the services start again. Sometimes starting up a service takes much longer than the actual run time of the workload; for time-critical workload, this may not be acceptable.

Standby services minimize the need to restart services when resources are allocated to an application by allowing these services to keep running. Standby services also allow other consumers to use these resources when there is no workload for the running service. This is due to the fact that standby services do not occupy slots, thereby allowing Platform EGO to allocate these resources to other applications. Once the service instance is associated with a slot and is used to run tasks, it is no longer considered a standby service.

Refer to *Using standby services to reduce service startup times* in the *Cluster and Application Management Guide*.

Security

Configurable roles & access rights (Phase 1)

Platform Symphony, out-of-the-box, allows a Platform EGO user account to be assigned any of the following roles: cluster administrator, consumer administrator, or consumer user. Each of these roles is associated with a fixed set of permissions that grant access to specific system controls and operations. For example, consumer users only have access and control over their own workload units and cannot access workload units of other consumers.

At some sites, cluster administrators/users and their functions may not map exactly to Platform Symphony's preconfigured security model for user roles. It may be desirable to have flexibility in the assignment of privileges. In this version of Platform Symphony, six new permissions have been created (additional permissions will be created in future releases), which can be applied to a role. The, role can be subsequently assigned to user(s).

Refer to *Customizing user roles* in the *Cluster and Application Management Guide*.

Active Directory Integration

Users can log into Platform Symphony using the same <domain>\<username> and password they use to log into a Windows machine in your organization. The cluster administrator can log into Platform Symphony by using the username 'Admin' and the password of the mapping account specified in adauth.conf.

Refer to to *Integrating Symphony with Active directory* in the *Cluster and Application Management Guide*.

Data Management

Multi-core Common Data Optimization

When common data for a session is sent to compute hosts, Platform Symphony, by default, sends one copy of the data to each SIM. In the case of one session running multiple SIMs on the same host, the same data will be sent multiple times. The potential drawback of having to send and maintain multiple copies of common data and common data updates is the burden it puts on host memory and network bandwidth.

This version of Platform Symphony allows you to optimize the distribution of common data and common data updates so that only one copy of the data is sent to each host serving the same session. The single copy of common data is shared between all SIMs and SIs associated with the same session on the same host.

Refer to *Optimizing common data for multi-slot hosts* in the *Cluster and Application Management Guide*.

Usability & Manageability Improvement

Configurable task filters

Time-based filters are provided for session/tasks in this version of Platform Symphony. These filters make searching for your tasks/session much easier. For example, you could ask Platform Symphony to list the tasks that finished between Monday Aug 4, 10 am and Monday Aug. 4, 11am. Or for example, you could ask the system to list those tasks that ran between the Monday Aug 4, 10 am to Monday Aug. 4, 11am on a certain compute host.

Configurable Paging Directory

In previous versions of Platform Symphony, the swapping of user data for recoverable and non-recoverable sessions was done at one location. For recoverable sessions, this location must be on a Shared File System for the SSM to recover in case of failure; however, for non-recoverable sessions the data does not need to on a Shared File system since it is swapped to disk to free up memory in the SSM. In Platform Symphony 5.0, the location for paging of the user data (input, output, common data and updates) for non-recoverable sessions can be separate from the location used for recoverable sessions. This can increase performance, task throughput and increase grid utilization.

Configurable JRE Home for Platform Management Console and PERF

Platform Management Console (PMC) and PERF can be configured to use a different JRE than the out of the box version.

Client Host Meta Info Display

In Platform Symphony 5.0, additional metadata relating to sessions is stored in session attributes and session history for troubleshooting purposes. The information allows you to identify which host/IP the work was submitted from, version of Symphony library used to submit workload, and OS user that was logged into the client machine to submit workload.

Debugability Enhancements - Policy Counter Display

There are many enhancements to improve debugability including additional scheduling counters. For example, you can view the number of slots demanded by the SSM from Platform EGO for the application, the number of slots each session deserves verses allocated, and details of how slots are distributed from each resource group to the sessions.

Additional Environment Support

- .Net 3.5 support, VS 2008
- Vista Support
- SuSe 10, Glibc 2.4
- Aix 5 and Solaris 64 SDK support
- On-demand support for additional platforms

For a complete list of supported platforms and compilers, refer to:

http://www.platform.com/Products/platform-symphony/symphony-system-requirements

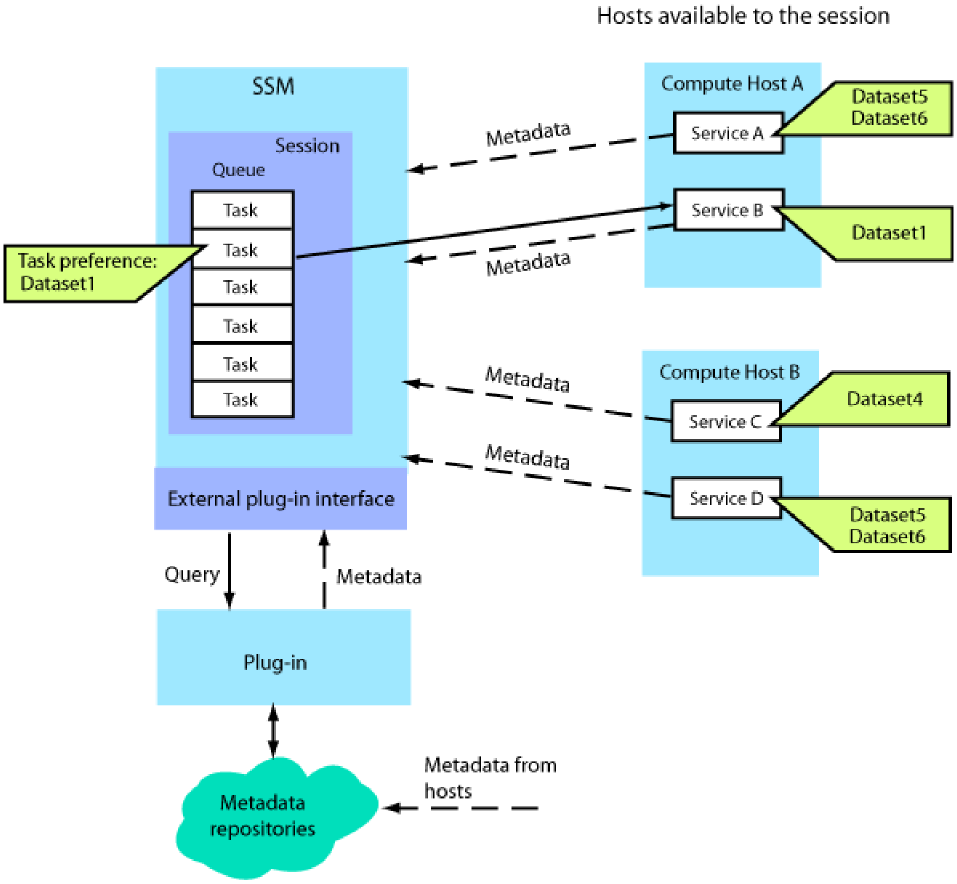
Platform Symphony Add-on Products

Data Affinity

Data Affinity reduces network traffic between hosts in the cluster and allows for more flexible scaling of data-intensive applications with improved performance. The Data Affinity add-on product contains the Data-aware Scheduling feature.

Data-aware scheduling allows Platform Symphony to intelligently schedule application tasks and improve performance by taking into account data location when dispatching the tasks. By directing tasks to resources that already contain the required data, application runtimes can be significantly reduced. As well, this feature can help to meet the challenges of latency requirements for real time applications.

The following example illustrates the concept of data-aware scheduling at the task level. The data preference expression has been evaluated and it is determined that the task prefers to run on a service instance where the service instance already possesses Dataset1. The SSM collects metadata (service attributes) from all the resources available to the session at that moment and selects Service B since it is the best match for the task according to the specified preference. The task is subsequently dispatched to Service B.



- Packaging: No additional package is required.
- Deployment: No additional deployment is required.
- Licensing: Additional software licensing is required for data-aware scheduling.

Refer to *Data-aware Scheduling User Guide*.

Multi-core Optimizer

The Multi-core Optimizer reduces capital and operating expenses by improving utilization of resources in mixed environments. It can also reduce I/O and memory contention in multi-core environments. Dynamic Service and Common Data Optimizer are two features of the multi-core optimizer add-on product.

Dynamic Service

The Dynamic Service feature allows dynamic control of service-to-slot mapping to optimize core, memory, and I/O use. This enables efficient running of workload made up of multi-threaded, single-threaded, I/O-intensive, and compute-intensive tasks from a single application and/or a mix of applications in a multi-core/heterogeneous environment.

- Packaging: No additional package is required.
- Deployment: No additional deployment is required.
- Licensing: Additional software licensing is required for the dynamic service feature.

Refer to *Dynamic Service User Guide*.

Common Data Optimizer

The Common Data Optimizer feature optimizes the distribution of common data and common data updates so that only one copy of the data is sent to each multi-core host serving the same session. The single copy of common data is shared between all SIMs and SIs associated with the same session on the same host.

Refer to the *Cluster and Application Management Guide*.

What’s Changed?

Messages and Logs

Error messages

The following messages are no longer logged by the SSM if the SSM intends to terminate the service instances.

```
2009-09-02 16:25:59.025 Eastern Standard Time INFO [2448:2472]
servers.common.resmngr.VEMResManager - VEMResManager::signalContainer() : Signal container
<575> with signum <7>
```

```
2009-09-02 16:25:59.025 Eastern Standard Time WARN [2448:2488]
servers.common.resmngr.VEMResManager - Domain <VEM,Application>: Service activity exits with
exception, container 631, reason: Terminated by SIGKILL, status 66304.
```

Application development

DE cluster host limit

The maximum number of hosts in a Symphony DE cluster is now limited to two.

Product licensing

A license audit log has been added to Symphony. The log records license usage over a period of time.

Resource management

Application-level ResourceGroupName

Refer to *Multiple Resource Group Selection at Application level* in the *What’s New* section.

taskLowWaterMark and taskHighWaterMark

The taskLowWaterMark parameter in the application profile is now limited to two settings: 0 and 1.

- If taskLowWaterMark is 0, SSM will not release any resources
- If taskLowWaterMark is 1, SSM will release resources when the resources cannot be used by current unfinished workload.

The value of taskHighWaterMark is now fixed at 1; this means that for every unprocessed task, 1 CPU slot is requested (assuming the session has a service-to-slot ratio of 1:1).

Security

EGO no longer supports the creation of user names containing ‘/’ or ‘\’.

Workload management

- Task dispatching order has changed if the resource preference feature or data-aware scheduling feature is enabled.
- Number of canceled tasks in an application has been added to PMC and CLI output.

Command changes

CLI

- CLI now supports the use of “” to pass empty values.
- CLI now supports the use of session name to query sessions.

egosh

- “egosh rg”, “egosh alloc list”, “egosh alloc view”, and “egosh consumer alloc” outputs have changed to support “Allocated” and “Occupied”.
- Resource group name, consumer name, host name, user name, and client name are truncated and '*' is appended if the name is too long to display in the output of "egosh" commands.

Character and value limitations

- The session type name in the application profile cannot exceed 128 characters.
- The reason/comment associated with killing a session or task cannot exceed 64 characters.
- The user name (owner) associated with a session cannot exceed 40 characters.
- The session name only supports alphanumeric, '-', and '_' characters. An environment variable has been introduced to preserve compatibility and allow session names with other characters; refer to SOAM_SESSION_NAME_COMPATIBILITY in the environment variables section of the Symphony Reference document.
- Maximum user-configurable time-to-live time changes to 10 years for a valid soamlogon session.

Configuration

The default value of EGO_DEFINE_NCPUS has changed from "procs" to "cores".

Known Issues

Antivirus software can block Symphony daemons

If you install Symphony on a host that is running antivirus software such as ESET NOD32, the following Symphony daemons may get quarantined, preventing the cluster from starting up.

- wsm
- wsgserver
- vemkd
- egosh
- pem

Restoring the quarantined daemons allows the cluster to start up.

Connector for Excel upgrade

If you have Symphony 5.0 installed, it is recommended that you deploy and use the 5.0 version of the Excel Connector service.

If you want to run version 4.x of the Excel Connector service in a 5.0 cluster, you must do one of the following:

- Copy the `com_platform_log4cxx_097_1.dll` file to the 5.0 `lib` directory.
- Repackage the 4.x Excel Connector service with `com_platform_log4cxx_097_1.dll` and redploy it.

Cannot open "Manage Service Packages" page on PMC after parallel install of Symphony 4.1 and 5.0 on the same host

If you install Symphony 5.0 after you install 4.1 on the same host, you may not be able to open the Manage Service Packages page on the Symphony 4.1 PMC. This is a known issue caused by a change in the SOAM_HOME environment variable during installation of Symphony 5.0. To resolve this issue, add SOAM_HOME in the `gui_service.xml` file under the relevant 4.1 configuration directory, i.e., share_top/gui/conf/, in each hostType section:

```
<ego:EnvironmentVariable name="SOAM_HOME">${EGO_TOP}\soam</ego:EnvironmentVariable>
```

Wildcard is not supported when querying tasks by task tag

The PMC and CLI do not support wildcard character "*" when querying tasks by task tag.

Migrating your .NET applications from a previous release of Symphony

A .NET application built using Symphony 4.1 or earlier versions will not automatically run in a Symphony 5.0 environment. This is a known issue caused by the .NET embedded "Publisher's Policy" infrastructure that is strictly enforced by the framework. Any published assembly with a version number that is different from your current Symphony release will be treated as incompatible by the framework. This incompatibility can be resolved in one of two ways:

1. Recompile your application using the Symphony 5.0 SDK. By recompiling, your application will automatically reference the new version of the assembly and will run without problems. This is the best option and is the strategy recommended by Microsoft for maximum compatibility.
2. Create a configuration file for your client and service to override the “Publisher’s Policy”. Since the Symphony 5.0 assembly is designed to be fully compatible with code using a previous version, it is safe to override the publisher’s policy and allow the framework to ignore the version number difference between Symphony 5.0 and your application’s dependence on a previous release of Symphony.

   The override can be achieved by creating a file named <complete executable filename>.config. For example, if your client executable has the name “SyncClient.exe” then you would need to create a file named "SyncClient.exe.config" and place it in the same location as the executable. The .NET framework will automatically parse the file and use it to override the "Publisher's Policy". Here is an example of the .config file contents:

   ```
   <?xml version ="1.0"?>
   <configuration>
     <runtime>
       <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
         <publisherPolicy apply="no"/>
         <dependentAssembly>
           <assemblyIdentity name="Platform.Symphony.Soam.Net"
            publicKeyToken="ebe3ab0caa75a162" culture="neutral" />
           <bindingRedirect oldVersion="4.1.0.0"
            newVersion="5.0.0.0"/>
         </dependentAssembly>
       </assemblyBinding>
     </runtime>
   </configuration>
   ```

   If your service is to run in the Symphony 5.0 environment, you need to add the .config file to the service package and redeploy the package.

Documentation Errata

R_MinimumServices

In the "Maintaining data affinity between a session and service instances" topic of the Cluster and Application Management Guide, the following configuration directive was incorrectly considered a best practice. Actually, overlapping resource group filters can be beneficial in some cases such as when using the dynamic service feature.

- Original text:

  "As a best practice, when configuring the R_MinimumServices policy with multiple session types for an application, ensure that each resource group name in the resource group filters is unique among all of the session types; otherwise you may get one less than the configured number of minimum services running."
- Updated text:

  "Note:When configuring the R_MinimumServices policy with multiple session types for an application, if a resource group name in the resource group filters is not unique among all of the session types, you may get one less than the configured number of minimum services running."

Delaying the release of slots after running workload

In the "Delaying the release of slots after running workload" topic of the Cluster and Application Management Guide, the following text was updated

- Original text:

  "After a session is suspended or killed, the service will wait for the slot release delay time to expire before the resources are released back to EGO."
- Updated text:

  "After a session is suspended or killed, if other outstanding workload does not use the service, the service will wait for the slot release delay time to expire before the resources are released back to EGO."

The following text was added to the third paragraph:

"Configuration for a service overrides the application-level configuration."

In the "Exclusive host" section, the following text was updated:

- Original text:

  When exclusive host is configured, an SSM must release all of the slots on the host back to EGO at once. All services on the host will be terminated (and the slots released) when the slot release delay time has expired for all of the services on that host.
- Updated text:

  When exclusive host is configured, an SSM must release all of the slots on the host back to EGO at once. All services on the host will be terminated (and the slots released) when all services are idle and the slot release delay time has expired for all of the services on that host.

FAQ about ICC8.1 compiler

The following topic was added to the "Application Configuration" FAQs:

Do I need any special settings to run my application if the application is compiled with ICC8.1?

The ICC8.1 library path should be added to the LD_LIBRARY_PATH environment variable of the client and the application profile."

Resource-aware Scheduling

The table of contents in the Cluster and Application Management Guide incorrectly refers to the "Resource-aware scheduling for sessions" topic as "Specifying resource preference for sessions and tasks".

egosh consumer alloc command

The Symphony Reference is missing the -g option for the egosh consumer alloc command. The -g option provides resource group details.