



## **Moving PowerBuilder® to EAServer—Tips and Tricks**

**Some technical hints that will help ensure success when moving PowerBuilder applications to EAServer.**

One of the things I wanted to do in this article was provide a set of tips and tricks for our EAServer users. This article is a subset of some of the tricks that I discussed during my Tips and Tricks presentation at this year's TechWave. I want to talk about 3 specific tricks that you could use to improve your performance. They are OS settings, caching, and how to reduce the amount of memory used by the PBVM within EAServer.

## OS Issues:

When you are using Solaris in order to maximize your EAServer performance you may have to make some modifications to the OS. The first of these settings will help avoid those nasty "connection refused" errors that can be received by clients' applications. To do this you may need to increase your max number of TCP connection requests. This is achieved by use of the `nnd` command. The `nnd` command's `get` and `set` options are used for selected configuration parameters in some kernel drivers. The two kernel drivers that we are interested in are `tcp_conn_req_max_q` and `tcp_conn_req_max_q0`.

- `tcp_conn_req_max_q` is the maximum number of completed connections waiting to return from an `accept` call as soon as the right process gets some CPU time.
- `tcp_conn_req_max_q0` is the maximum number of connections with handshake incomplete.

To put this in non-techno-speak, the first specifies maximum length of the *completed connection queue*, the second the size of the *incomplete connection queue*. These are important because if they are too small Solaris may refuse the connection before they even get to EAServer.

To get the current values enter the command:

```
nnd -get /dev/tcp tcp_conn_req_max_q
nnd -get /dev/tcp tcp_conn_req_max_q0
```

These values should be set to at least 1024. To set the values use the following 2 commands:

```
nnd -set /dev/tcp tcp_conn_req_max_q 1024
nnd -set /dev/tcp tcp_conn_req_max_q0 1024
```

Please note that there is no reboot required after setting the command. As part of this change a new parameter has been added. The parameter is `com.sybase.jaguar.listener.solaris.tli.maxoutcon`. This value should match the value of `tcp_conn_req_max_q` and `tcp_conn_req_max_q0`. The default value is 128 which is also the default values for `tcp_conn_req_max_q` and `tcp_conn_req_max_q0`. Also, you should note that there can be significant memory increase when this value increases. So you may have to tune these values.

Another parameter that can help you is to increase the number of allowable open files. This is because we open file handles when processing objects. To be sure we have enough file handles and to ensure that EAServer does not either refuse a connection or increase wait time you may need to increase your file handles. To do this edit the `/etc/system` file and be sure that `rlim_fd_cur` and `rlim_fd_max` are set to at least 4096. It should be noted that if you do this you would have to reboot your Solaris box.

## Caching

With caching there is really one objective—to improve your performance. This can be done either by not having to go to the database, the file system, or re-run some piece of business logic. Within EAServer there are a number of ways to improve performance using caching.

Caching can be done to either components, JSPs or Servlets. For components the intent of object caching is to keep from having to read from the database. For update transactions, you may

be able to avoid any SQL “select” statements, while still having to execute a SQL “update” statement. For read-only transactions, you may be able to entirely avoid the execution of SQL statements. And for CMP entity finder methods that return collections, you may be able to run one query instead of N+1 queries for finders returning a collection containing N entities. Your performance gains will depend on:

- The frequency of re-use of objects with same key (e.g. how often does a new transaction use the same entity as a previous transaction for which an entry is still present in the object cache);
- The ratio of update to read-only transactions;
- The size of data set (e.g. database table);
- The size of object cache and the configured timeout for cache entries.

To set up caching at the component level you need to use Jaguar Manager. Cache setup is done on the Persistence tab of the component.

Servlet and JSP caching allows dynamic HTML content to be cached by the server. This can drastically improve the speed at which HTML content is served to the user; the server does not have to do any additional processing in order to return a result to the user. In our tests in the lab we have seen an increase of up to 60 times faster with caching turned on.

JSP and Servlet caching can use either simple or object caching. Simple cache implements Java side Servlet cache that will cache outputs of Servlet/JSP in java core memory. This provides for very fast response for Servlets or JSPs for which the output will not be changed during the timeout period. To turn this feature on simply set `com.sybase.jaguar.servlet.javacache.enabled` to true. The default is set to false. If you are using sessions you may need to modify the setting for `com.sybase.jaguar.sevlet.javacache.session`. The settings are:

- Keep—If request includes session id, we'll try to get the session.
- Create—If request includes session, we'll try to get the session.
- No—Do not check or return “Set-Cookie” session header.

JSP and Servlet object caching uses the same caching mechanism as component object caching. It also provides for a more refined caching algorithm. Which mechanism you use will depend on how refined you want your caching to be. For more information on these two features please consult the *EAServer Programmers guide* (chapters 20 and 25).

## Reducing PBVM memory size

One of the biggest issues is how to reduce the amount of memory the PBVM takes at runtime. In some cases the amount of memory used by the PBVM can become very large. Some of this can be attributed to the way that PB Components are deployed to EAServer. Let me give you an example, and then tell you how to fix the problem. Let's say you have a small application that consists of 3 PBLs and has 5 components. When you deploy the components to EAServer, a directory structure like the following is created:

```
%JAGUAR%  
  Repository/  
    Component/  
      <Package Name>
```

Where %JAGUAR% is your EAServer installation directory. Within the Package subdirectory you will find a property file for each component that was deployed. This file has a field in it called `com.sybase.jaguar.component.pb.cookie`. The cookie is incremented every time the component is deployed. This incremented cookie then points to the directory under the

## International Contacts

<b>Argentina</b> +5411 4313 4488	<b>Korea</b> +82 2 3451 5200
<b>Australia</b> +612 9936 8800	<b>Malaysia</b> +603 2142 4218
<b>Austria</b> +43 1 504 8510	<b>Mexico</b> +52 5282 8000
<b>Belgium</b> +32 2 713 15 03	<b>Netherlands</b> +31 20 346 9290
<b>Brazil</b> +5511 3046 7388	<b>New Zealand</b> +64 4473 3661
<b>Bulgaria</b> +359 2 986 1287	<b>Nigeria</b> +234 12 62 5120
<b>Canada</b> +905 273 8500	<b>Norway</b> +47 231 621 45
<b>Central America</b> +506 204 7151	<b>Panama</b> +507 263 4349
<b>Chile</b> +56 2 330 6700	<b>Peru</b> +51 11 221 4190
<b>China</b> +8610 6856 8488	<b>Philippines</b> +632 750 2550
<b>Colombia</b> +57 1 218 8266	<b>Poland</b> +48 22 844 55 55
<b>Croatia</b> +385 42 33 1812	<b>Portugal</b> +351 21 424 6710
<b>Czech Republic</b> +420 2 24 31 08 08	<b>Puerto Rico</b> +787 289 7895
<b>Denmark</b> +45 3927 7913	<b>Romania</b> +40 1 231 08 70
<b>Ecuador</b> +59 322 508 593	<b>Russian Federation</b> +7 095 797 4774
<b>El Salvador</b> +503 245 1128	<b>Singapore</b> +65 370 5100
<b>Finland</b> +358 9 7250 200	<b>Slovak Republic</b> +421 26 478 2281
<b>France</b> +33 1 41 91 96 80	<b>Slovenia</b> +385 42 33 1812
<b>Germany</b> +49 69 9508 6182	<b>South Africa</b> +27 11 804 3740
<b>Greece</b> +30 1 98 89 300	<b>South Korea</b> +82 2 3451 5200
<b>Guatemala</b> +502 366 4348	<b>Spain</b> +34 91 749 7605
<b>Honduras</b> +504 239 5483	<b>Sweden</b> +46 8 587 70433
<b>Hong Kong</b> +852 2506 6000	<b>Switzerland</b> +41 1 800 9220
<b>Hungary</b> +36 22 517 631	<b>Taiwan</b> +886 2 2715 6000
<b>India</b> +91 22 655 0258	<b>Thailand</b> +662 618 8638
<b>Indonesia</b> +62 21 526 7690	<b>Turkey</b> +90 212 284 8339
<b>Israel</b> +972 3 548 3555	<b>Ukraine</b> +380 44 227 3230
<b>Italy</b> +39 02 696 820 64	<b>United Arab Emirates</b> +971 2 627 5911
<b>Ivory Coast</b> +225 22 43 73 73	<b>United Kingdom</b> +44 870 240 2255
<b>Japan</b> +81 3 5210 6000	<b>Venezuela</b> +58 212 267 5670
<b>Kazakstan</b> +7 3272 64 1566	<b>Asian Solutions Center</b> +852 2506 8700

For other Europe, Middle East, or Africa inquiries:  
+33 1 41 90 41 64 (Sybase Europe)

For other Asia Pacific inquiries:  
+852 2506 6000 (Hong Kong)

For other Latin America inquiries:  
+305 671 1020 (Miami)

component where the actual PBD lives at runtime. That directory structure looks like the following:

*<Component Name>*

*C<n>*

*<one or more PBDs implementing component>*

Within EAServer, the PBVM has a unique instance of a class group loader assigned to each distinct occurrence of a library list. That library list is based on the component property `com.sybase.jaguar.component.pb.librarylist`. So in our example, that value would be something like `$PBD1.PBD;$PBD2.PBD;$PBD3.PBD;...` where the `$` indicates the relative path of the component. So for each component there is a unique set of libraries in memory. In a development environment this is a good thing because it provides for great flexibility. In a production environment, however, this will cause much larger memory usage because there is a unique library list for each PB component deployed because of the substitution that occurs. To correct this problem the following steps should be taken.

1. Include all PBLs containing the NVOs that are sharing resources in a single, combined library list.
2. Deploy each component, as you normally would, just ensure you use the same deployment options in terms of PBL consolidation.
3. Copy the generated PBDs to a common location on your production server.
4. Use jaguar manager to modify the `com.sybase.jaguar.component.pb.librarylist` property in the properties file of each component to be the exact same list by referencing a fully qualified path to the common location of your PBDs.
5. Refresh the components.
6. Backup the changes to the components' properties files.

This will help reduce the amount of memory used by the PBVM at start time.

Hopefully you can take advantage of some of these tricks and tips to improve the performance and memory utilization of your EAServer environment.



Sybase, Inc.

Worldwide Headquarters

5000 Hacienda Drive  
Dublin, CA 94568-7902 USA  
Tel: +800 8 SYBASE  
[www.sybase.com](http://www.sybase.com)