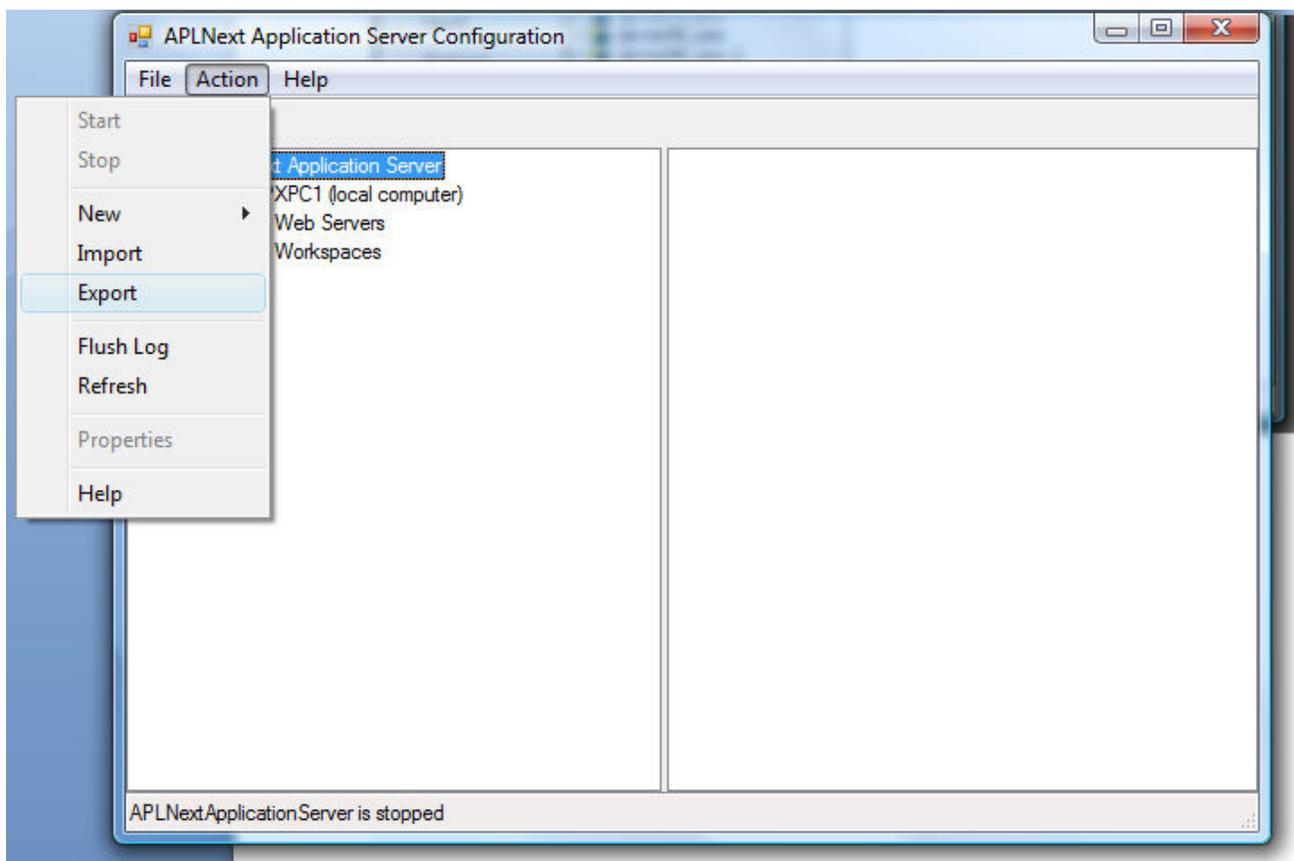


APLNextApplicationServer Importing and Exporting Configurations

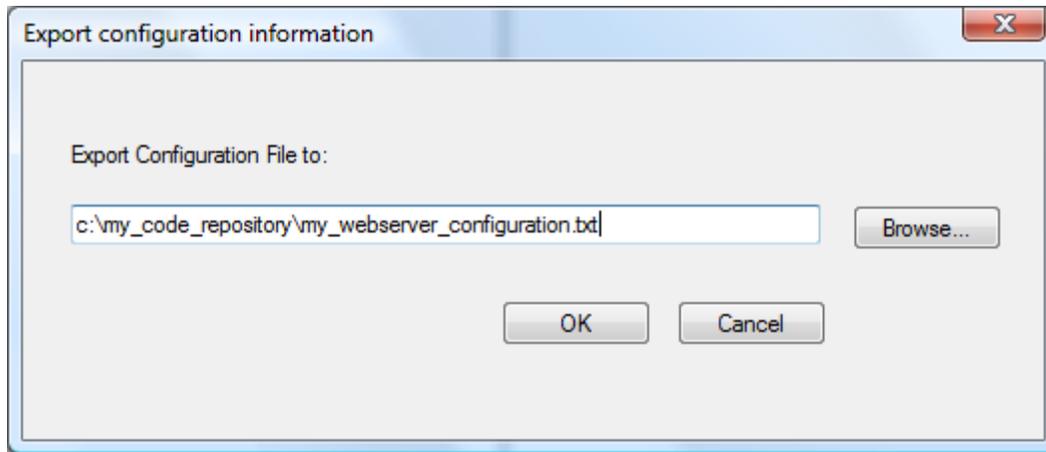
It is suggested that the programmer-developed APLNextApplicationServer configuration of web servers and virtual workspaces be saved using the configuration export action. APLNextApplicationServer configurations may be imported using the inverse action provided by the product.

Note that when an APLNextApplicationServer configuration file is imported, it may be necessary to restart the APLNextApplicationServer servers. This can be done either by re-starting the machine on which the APLNextApplicationServer software has been installed or may be done manually using the 'APLNextApplicationServer Admin' tool.

To begin the export action, open the 'APLNextApplicationServer Adm' tool, in the left pane of the dialog select on the 'APLNext Application Server' node and click the 'Action' menu item and click the 'Export' option on the pop-up menu.



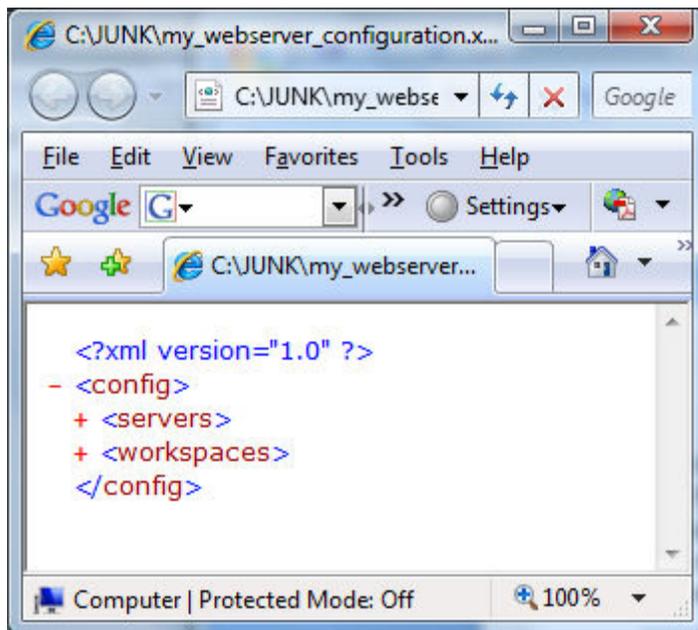
The 'Export configuration information' dialog will be presented. Enter the full Windows file system path and name of the file to contain your APLNextApplicationServer configuration or use the 'Browse' button to present the Microsoft Windows file save as dialog to point to the desired path and filename. Click the 'Ok' button on the 'Export configuration information' dialog to create the exported configuration file.



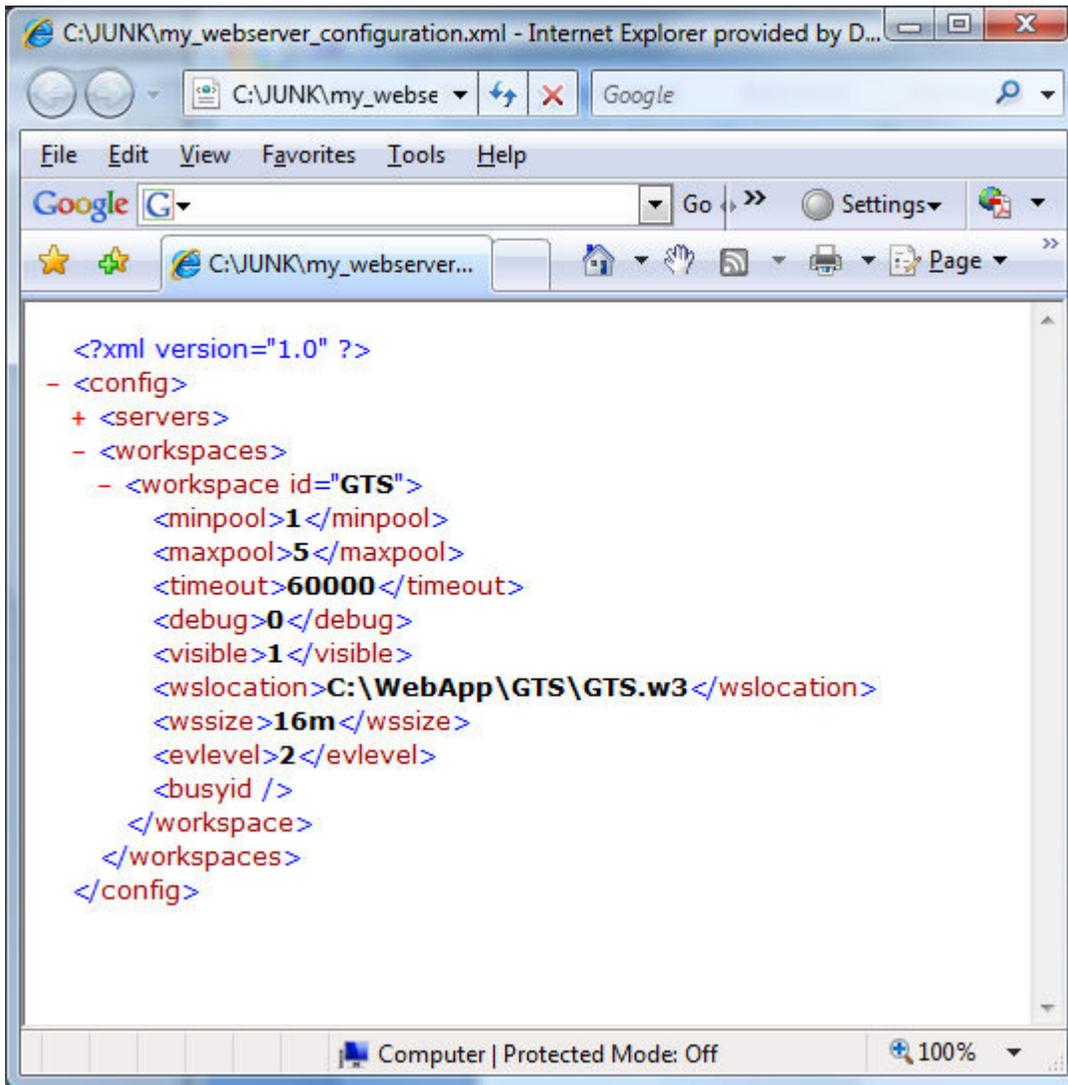
The resulting file is an xml-format file, so an extension of .xml or .txt may be used. If the .xml file extension is used, the exported configuration file may be viewed using Microsoft Internet Explorer.

Any editing tool for xml-format files may be used to view and modify the contents of the exported configuration file.

The xml-format configuration file is divided into two main tags, 'servers' and 'workspaces', which correspond to the display of this information in the 'APLNextApplicationServer Admin' tool main form. In fact all of the information displayed by the 'APLNextApplicationServer Admin' tool is contained in the exported configuration file.



A typical 'workspace' sub-tag element in the xml-format configuration file looks like:



The screenshot shows an Internet Explorer browser window displaying an XML configuration file. The address bar shows the file path: C:\JUNK\my_webserver_configuration.xml. The XML content is as follows:

```
<?xml version="1.0" ?>
- <config>
+ <servers>
- <workspaces>
  - <workspace id="GTS">
    <minpool>1</minpool>
    <maxpool>5</maxpool>
    <timeout>60000</timeout>
    <debug>0</debug>
    <visible>1</visible>
    <wslocation>C:\WebApp\GTS\GTS.w3</wslocation>
    <wssize>16m</wssize>
    <evlevel>2</evlevel>
    <busyid />
  </workspace>
</workspaces>
</config>
```

A typical 'server' sub-tag element in the xml-format configuration file looks like:

C:\JUNK\my_webserver_configuration.xml - Internet Explorer provided by Dell

C:\JUNK\my_webserver_configuratio Google

File Edit View Favorites Tools Help

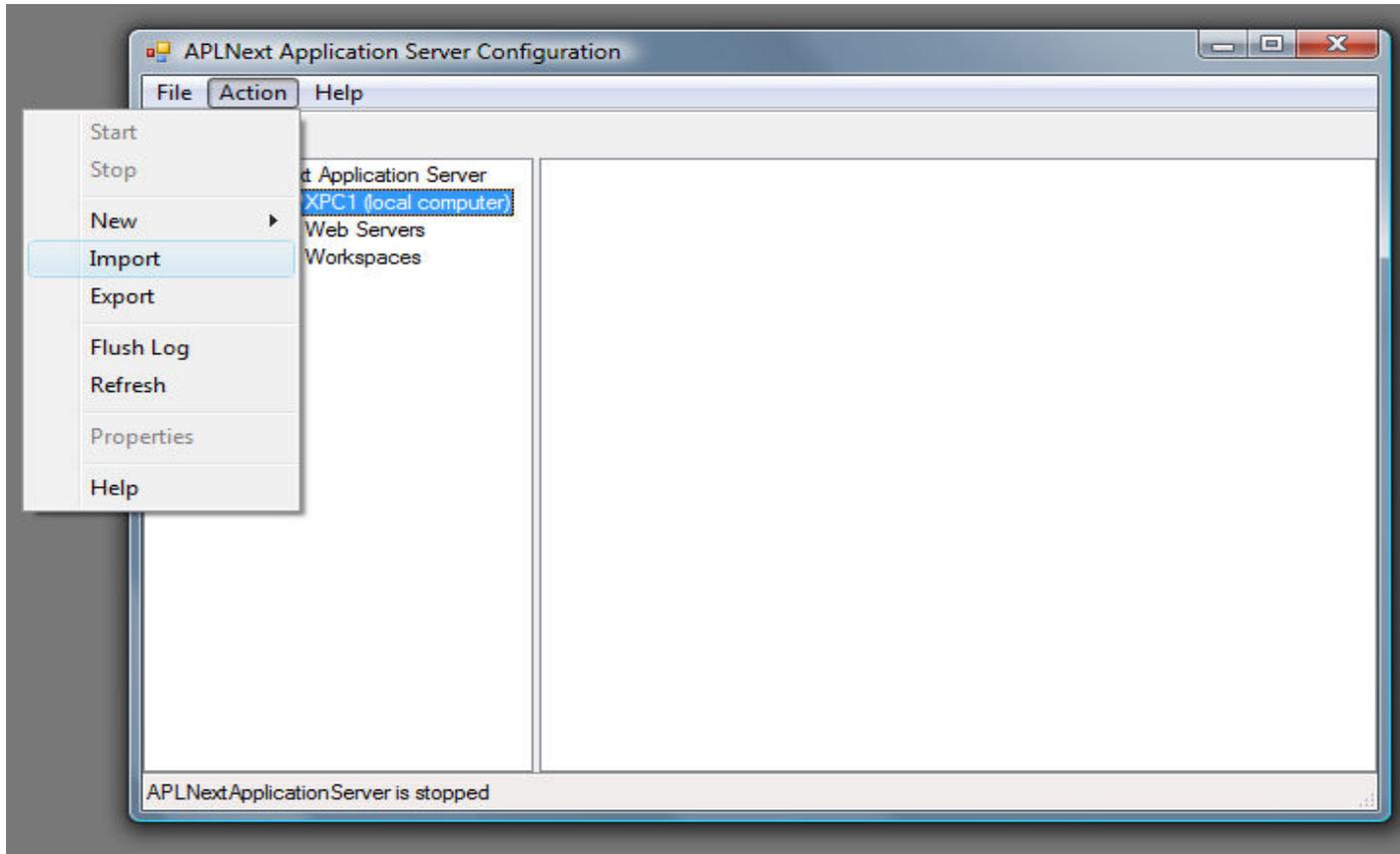
Google G Go Settings

C:\JUNK\my_webserver_conf... Page Tools

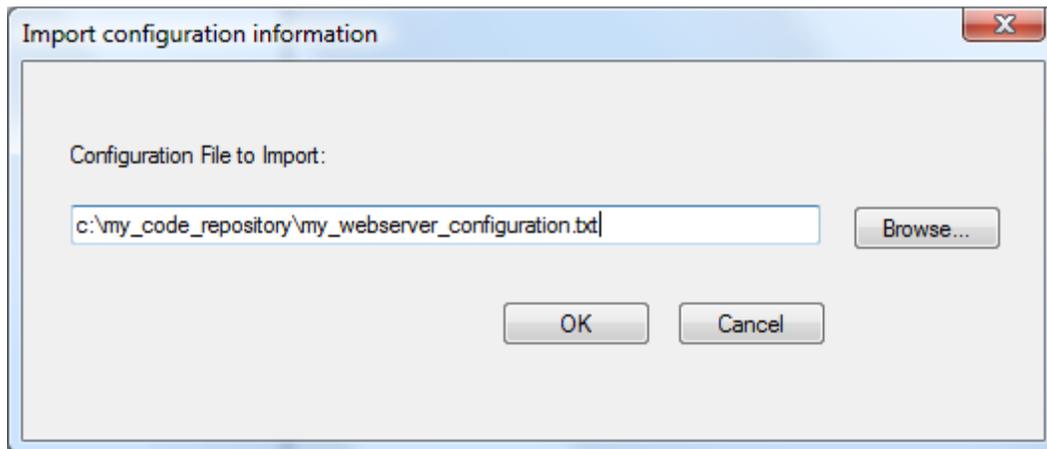
```
<?xml version="1.0" ?>
- <config>
- <servers>
- <server id="gts">
  <host>localhost</host>
  <port>9090</port>
  <publichttpdir>C:\WebApp\GTS</publichttpdir>
  <defaultfile />
  <state>started</state>
  <enable-default-file>False</enable-default-file>
  <connection-timeout>900</connection-timeout>
  <http-keep-alives>False</http-keep-alives>
  <enable-logging>True</enable-logging>
  <logfile-directory>C:\WebApp\GTS\Log</logfile-directory>
  <enable-content-expiration>False</enable-content-expiration>
  <expire-content>after;1;Hour(s)</expire-content>
  <custom-error name="400">Default</custom-error>
  <custom-error name="403">Default</custom-error>
  <custom-error name="404">Default</custom-error>
  <custom-error name="500">Default</custom-error>
  <custom-error name="501">Default</custom-error>
  <custom-error name="503">Default</custom-error>
  <custom-error name="504">Default</custom-error>
- <virtualpath name="/gts/ras090_save">
  <access />
  <wsid>GTS</wsid>
  <function>RAS090_SAVE</function>
  <rarg type="binarywrapl">RARG</rarg>
  <result type="apl2binarywrapl">r</result>
</virtualpath>
+ <virtualpath name="/gts/ras090_open">
+ <virtualpath name="/gts/ras090_open_2">
+ <virtualpath name="/gts/ui048_bssi_signin_sys">
+ <virtualpath name="/gts/ui048_bssi_signout_sys">
+ <virtualpath name="/gts/ui048_bssi_issessionexist_sys">
+ <virtualpath name="/gts/run_billing_tx">
+ <virtualpath name="/gts/ui048_bssi_permissions">
</server>
+ <server id="defaultwebsite">
</servers>
+ <workspaces>
</config>
```

Computer | Protected Mode: Off 100%

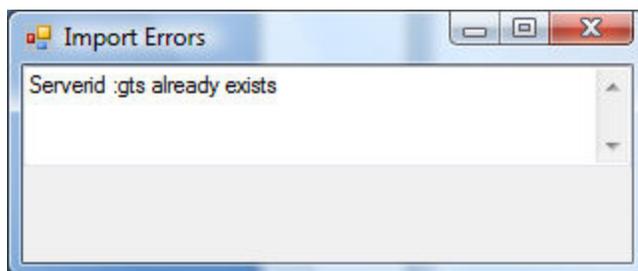
To begin the import action, open the 'APLNextApplicationServer Adm' tool, in the left pane of the dialog select on the 'APLNext Application Server' node, click the 'Action' menu item and click the 'Import' option on the pop-up menu.



The 'import configuration information' dialog will be presented. Enter the full Windows file system path and name of the file which contain your previously-exported and possibly-edited APLNextApplicationServer configuration file or use the 'Browse' button to present the Microsoft Windows file open dialog to point to the desired path and filename. Click the 'Ok' button on the 'Import configuration information' dialog to import configuration file.



When importing a configuration file, the APLNextApplicationServer Import action will check for conflicts between the servers and workspaces in the current active configuration and those in the configuration file to be imported. For example:



To resolve these conflicts use the 'APLNextApplicationServer Admin' tool to delete the conflicting elements from the current active configuration or use an xml-format file editor to delete the conflicting elements from the configuration file to be imported and then try the Import configuration action again.