



Intel® Smart Connect Technology Remote Wake with WakeMyPC

January 2014 - Revision 1.5

CDI/IBP #: 514349

Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm>.

Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

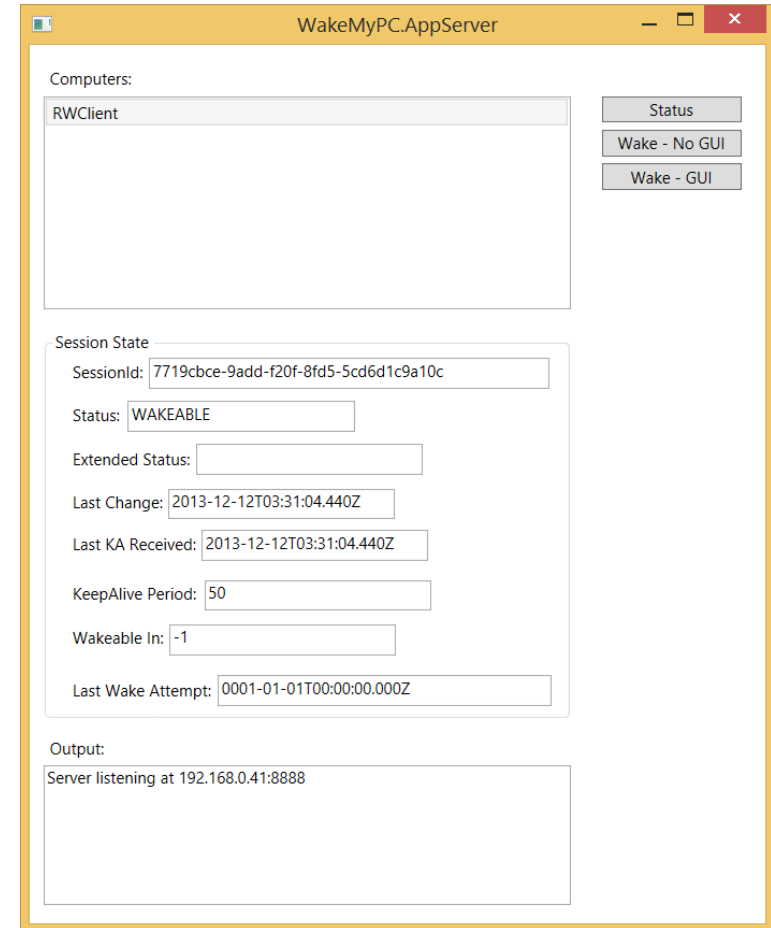
*Other names and brands may be claimed as the property of others.

Copyright © 2013-14, Intel Corporation. All rights reserved.



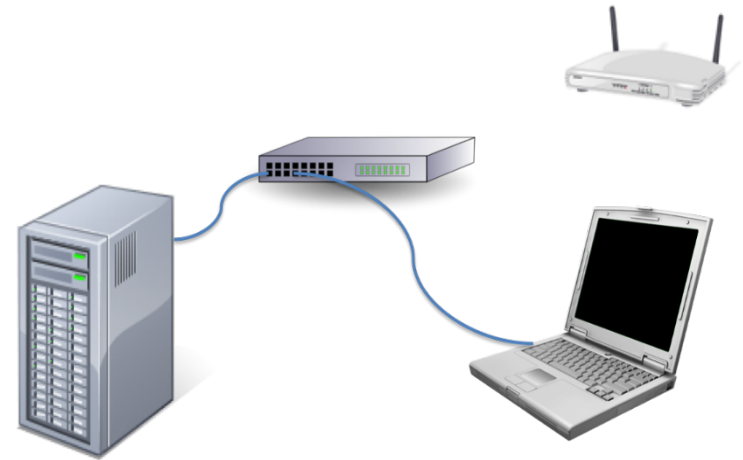
What is WakeMyPC?

- WakeMyPC is a client/server application for testing Remote Wake platform enablement
- It's free to use for anyone
- Development is ongoing
- Server application supports multiple client platforms



WakeMyPC Requirements

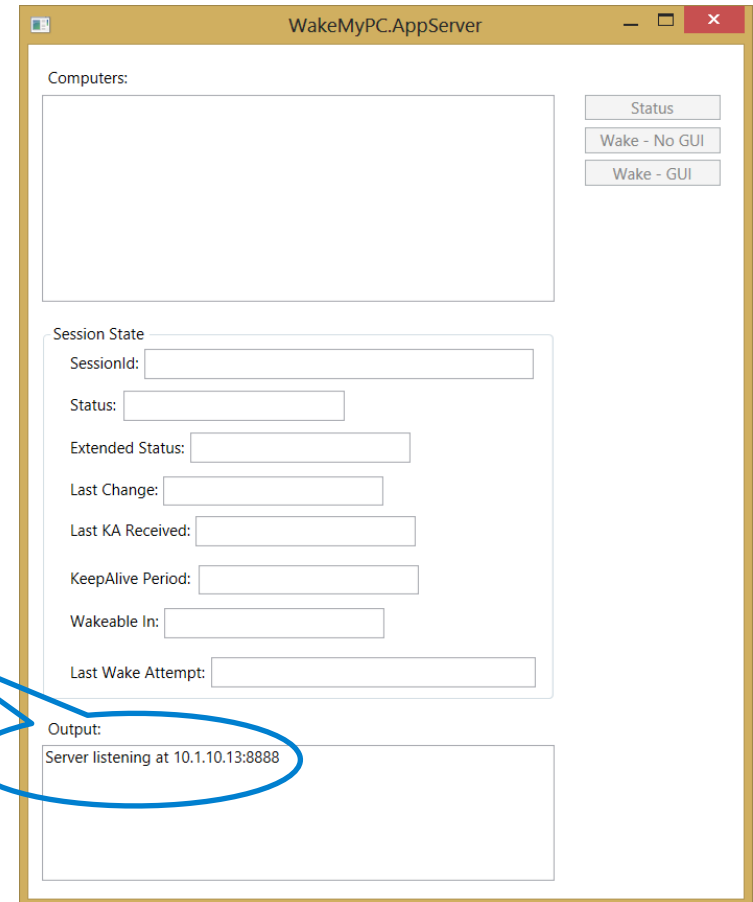
- Two platforms:
 - Server: Windows* 7 SP/Windows* 8/ Windows* 8.1 platform (LAN or WLAN)
 - Remote Wake Client: Windows* 8/ Windows* 8.1 platform with Remote Wake Enabled LAN and or WLAN
 - Windows* Firewall disabled on both platforms
 - Internet Connection with no proxy server(s)



WakeMyPC Server

On the server platform:

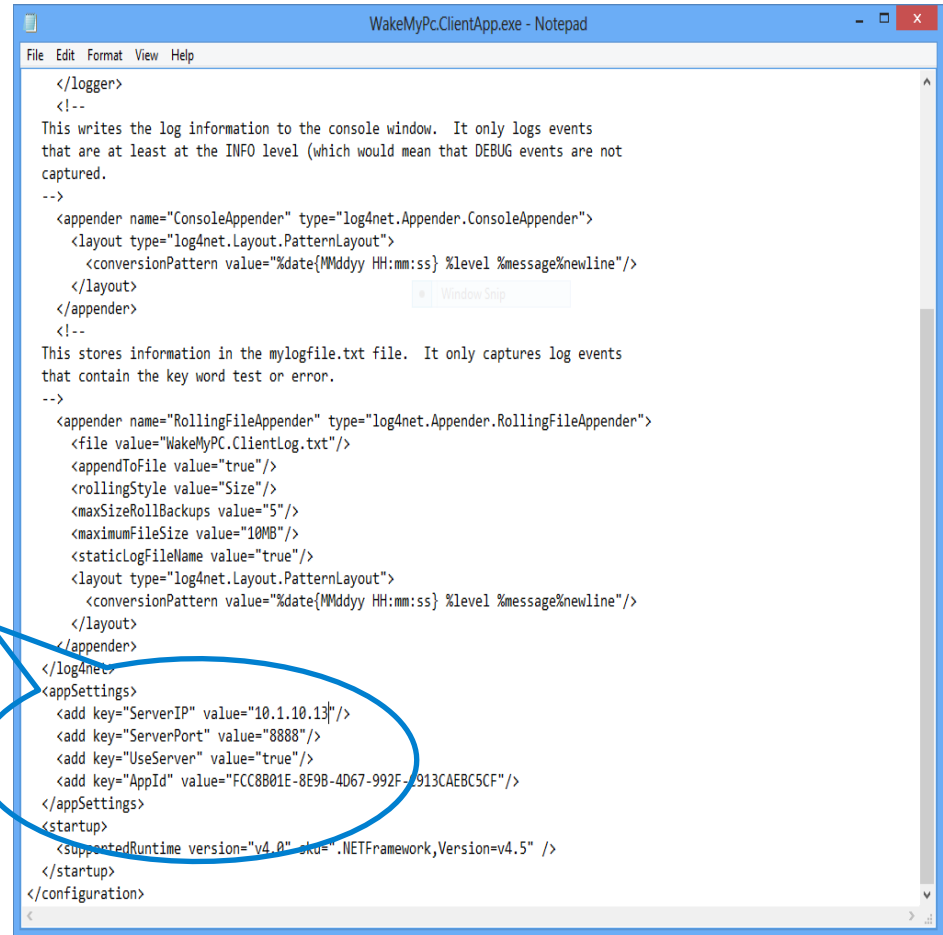
- Change directory to the WakeMyPC.Appserver
- Invoke the application WakeMyPC.AppServer.exe
- Application will print in the *Output* text box the IP address being used (this address as it will be required for the WakeMyPC Client application)



WakeMyPC Client

On the Remote Wake Client Platform:

- Change directory to the WakeMyPC.ClientApp
- Edit the file: WakeMyPc.ClientApp.exe.config
- At the bottom of the file in the "<appSettings>" section, change the value of the "ServerIP" to match the IP Address displayed in the in the server application
- Save the file



```
</logger>
<!--
This writes the log information to the console window. It only logs events
that are at least at the INFO level (which would mean that DEBUG events are not
captured.
-->
<appender name="ConsoleAppender" type="log4net.Appender.ConsoleAppender">
  <layout type="log4net.Layout.PatternLayout">
    <conversionPattern value="%date{MMddyy HH:mm:ss} %level %message%newline"/>
  </layout>
</appender>
<!--
This stores information in the mylogfile.txt file. It only captures log events
that contain the key word test or error.
-->
<appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
  <file value="WakeMyPC.ClientLog.txt"/>
  <appendToFile value="true"/>
  <rollingStyle value="Size"/>
  <maxSizeRollBackups value="5"/>
  <maximumFileSize value="10MB"/>
  <staticLogFileName value="true"/>
  <layout type="log4net.Layout.PatternLayout">
    <conversionPattern value="%date{MMddyy HH:mm:ss} %level %message%newline"/>
  </layout>
</appender>
</log4net>
<appSettings>
  <add key="ServerIP" value="10.1.10.13"/>
  <add key="ServerPort" value="8888"/>
  <add key="UseServer" value="true"/>
  <add key="AppId" value="FCC8B01E-8E9B-4D67-992F-913CAEBC5CF"/>
</appSettings>
<startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
</startup>
</configuration>
```



WakeMyPC Client

On the Remote Wake Client Platform:

- Change directory to the WakeMyPC.ClientApp
- Invoke the file: WakeMyPC.ClientApp.exe
- Upon invocation, the application will display the current session information and send this to the WakeMyPC.AppServer.
- When the WakeMyPC.AppServer receives the session information, it will display the computer name, SessionId and enable the "Status" button

The screenshot shows the 'WakeMyPc.ClientApp' window. It contains several input fields and buttons. A blue oval highlights the 'Messages' section, which displays 'WakeMyPC.Client Started...' and 'Sending SessionInfo to WakeMyPC.AppServer...'. A blue arrow points from this oval to the 'Status' button in the 'WakeMyPC.AppServer' window below.

WakeMyPc.ClientApp

Session Info

SessionId: 1c1979bc-f7a9-0f3d-320d-c9a11a1425bb

Wake URI: https://wr-1us.smartconnect.intel.com/sessions

Wake State: AVAILABLE Platform Type: mobile

LAN Status: NOT_PRESENT WLAN Status: AVAILABLE

Last Wake Packet: <none>

Hibernate Supported: True SubStatus: NO_SUBSTATUS

Context Info

S0 Duration (secs): Sx Duration (secs):

Messages

WakeMyPC.Client Started...
Sending SessionInfo to WakeMyPC.AppServer...

Sleep

The screenshot shows the 'WakeMyPC.AppServer' window. It displays a list of computers, session state information, and buttons for waking the computer. A blue oval highlights the 'Session State' section, which shows the 'SessionId' and 'Status' fields. A blue arrow points from this oval to the 'Status' button in the 'WakeMyPC.ClientApp' window above.

WakeMyPC.AppServer

Computers:

rwtest_1

Status

Wake - No GUI

Wake - GUI

Session State

SessionId: 1c1979bc-f7a9-0f3d-320d-c9a11a1425bb

Status:

Extended Status:

Last Change:

Last KA Received:

KeepAlive Period:

Wakeable In:

Last Wake Attempt:

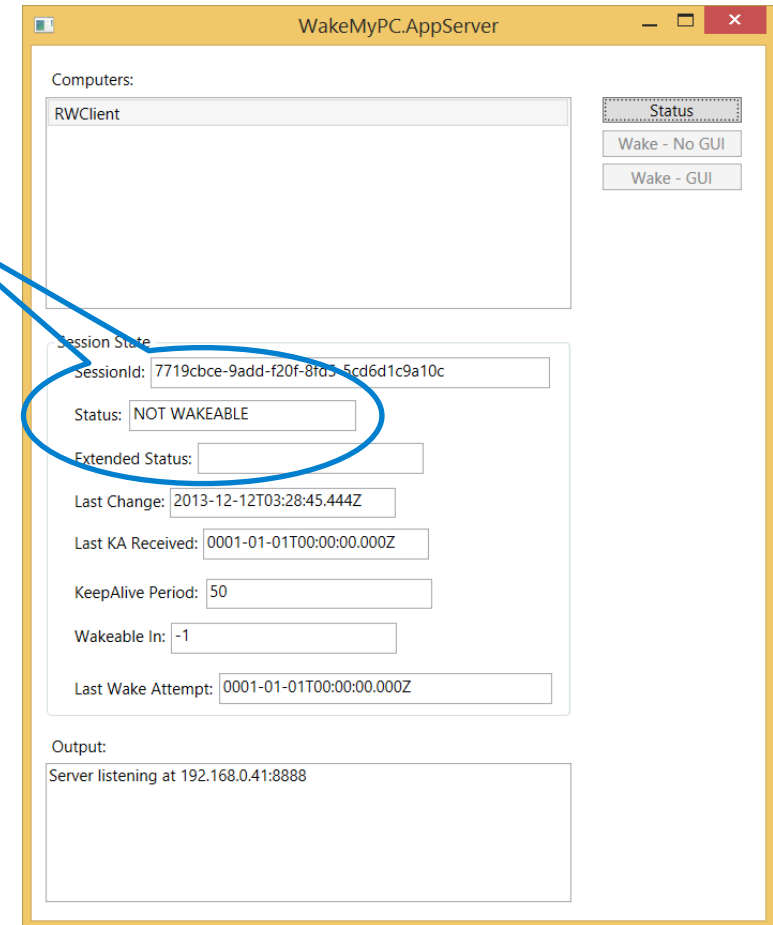
Output:

Server listening at 10.1.10.13:8888

WakeMyPC Server

On the Server Platform:

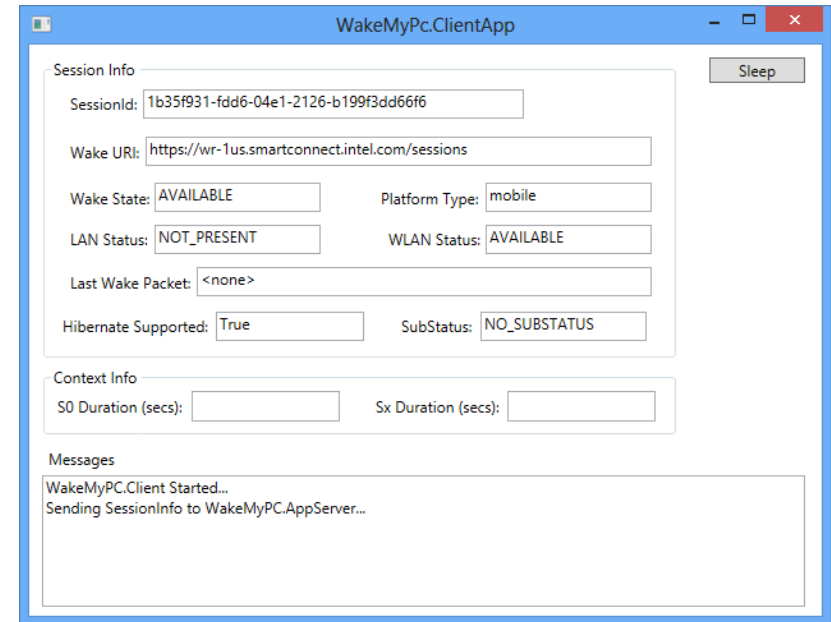
- Click the "Status" button and verify in the Session State information that the Client platform status is "NOT WAKEABLE". This state indicates that the Client platform is not in the sleeping state and thus cannot be remote waked.



WakeMyPC Client

On the Remote Wake Client Platform:

- Click the “Sleep” button to place the platform into S3 sleep state
- If OS Hibernate is supported, use the OS Power control to place the platform into the Hibernate state



The screenshot shows the WakeMyPc.ClientApp window with the following fields and values:

Session Info	
SessionId:	1b35f931-fdd6-04e1-2126-b199f3dd66f6
Wake URI:	https://wr-1us.smartconnect.intel.com/sessions
Wake State:	AVAILABLE
Platform Type:	mobile
LAN Status:	NOT_PRESENT
WLAN Status:	AVAILABLE
Last Wake Packet:	<none>
Hibernate Supported:	True
SubStatus:	NO_SUBSTATUS

Context Info	
S0 Duration (secs):	
Sx Duration (secs):	

Messages

WakeMyPC.Client Started...
Sending SessionInfo to WakeMyPC.AppServer...

WakeMyPC Server - S3

On the Server Platform:

- Once the Client platform is in the sleep state for several minutes click the "Status" button to check the Status of the Client platform. When the Status field of the Session State reports "WAKEABLE" and the field "Wakeable In" has a value of "-1" the two buttons under the Status button will be enabled to wake the platform.

Note:

It may take a few minutes for the Client platform and the Intel Wake Server to synchronize the platforms sleep state

If the field "Wakeable In" has a value of greater than "0", this indicates that the platform is not wakeable for that value. Click the "Status" button after waiting the time shown in the field.

The screenshot shows the WakeMyPC.AppServer web interface. The title bar is orange and says "WakeMyPC.AppServer". The main content area is white. At the top, there's a section labeled "Computers:" with a table showing "RWClient". To the right of the table are three buttons: "Status", "Wake - No GUI", and "Wake - GUI". Below the "Computers:" section is a section labeled "Session State" with several fields: "SessionId:" (7719cbce-9add-f20f-8fd5-5cd6d1c9a10c), "Status:" (WAKEABLE), "Extended Status:" (empty), "Last Change:" (2013-12-12T03:31:04.440Z), "Last KA Received:" (2013-12-12T03:31:04.440Z), "KeepAlive Period:" (50), "Wakeable In:" (-1), and "Last Wake Attempt:" (0001-01-01T00:00:00.000Z). At the bottom, there's a section labeled "Output:" with a text box showing "Server listening at 192.168.0.41:8888".

WakeMyPC Server - OS S4

To test Remote Wake from OS S4 (if supported on the client platform), the same steps are followed in the testing of the Remote Wake from S3, except the platform is manually put into Hibernate instead of Sleep.

The WakeMyPC.ClientApp will indicate if Hibernate wake is supported in the "Hibernate Supported" field. A value of "False" indicates that either a BIOS power-on password or disk password is enabled and thus no wake is supported.

The screenshot shows the WakeMyPC.ClientApp interface with the following fields:

- Session Info:**
 - SessionId: a996a329-755a-b824-7566-aedfa56c5448
 - Wake URI: https://wr-1us.smartconnect.intel.com/sessions
 - Wake State: AVAILABLE
 - Platform Type: mobile
 - LAN Status: NOT_PRESENT
 - WLAN Status: AVAILABLE
 - Last Wake Packet: <none>
 - Hibernate Supported: True
 - SubStatus: NO_SUBSTATUS
- Context Info:**
 - S0 Duration (secs):
 - Sx Duration (secs):
- Messages:**
 - WakeMyPC.Client Started...
 - Sending SessionInfo to WakeMyPC.AppServer...
 - Sending SessionInfo to WakeMyPC.AppServer...

A "Sleep" button is located in the top right corner.

The screenshot shows the WakeMyPC.ClientApp interface with the following fields:

- Session Info:**
 - SessionId: cdd580f3-d95e-24f6-f6b0-ab2f05f1f202
 - Wake URI: https://wr-1us.smartconnect.intel.com/sessions
 - Wake State: AVAILABLE
 - Platform Type: mobile
 - LAN Status: NOT_PRESENT
 - WLAN Status: AVAILABLE
 - Last Wake Packet: <none>
 - Hibernate Supported: False
 - SubStatus: NO_SUBSTATUS
- Context Info:**
 - S0 Duration (secs):
 - Sx Duration (secs):
- Messages:**
 - WakeMyPC.Client Started...
 - Sending SessionInfo to WakeMyPC.AppServer...
 - Sending SessionInfo to WakeMyPC.AppServer...

A "Sleep" button is located in the top right corner.

WakeMyPC Server

On the Server Platform:

- Two wake requests are now available:
 - Click the “Wake - No GUI” button to send a wake request to the Client platform to have it wake and keep the display off and audio muted (no sound heard) and after a few minutes go back to sleep state. You will see a new SessionId displayed after several seconds from when the platform resumes from sleep state.
 - Click the “Wake - GUI” button to send a wake request to the Client platform to have it wake and turn on the display.

The screenshot shows the 'WakeMyPC.AppServer' application window. It features a 'Computers' section with a list containing 'RWClient'. To the right of this list are three buttons: 'Status', 'Wake - No GUI', and 'Wake - GUI'. Below the 'Computers' section is the 'Session State' section, which displays various fields: 'SessionId' (7719cbce-9add-f20f-8fd5-5cd6d1c9a10c), 'Status' (WAKEABLE), 'Extended Status' (empty), 'Last Change' (2013-12-12T03:31:04.440Z), 'Last KA Received' (2013-12-12T03:31:04.440Z), 'KeepAlive Period' (50), 'Wakeable In' (-1), and 'Last Wake Attempt' (0001-01-01T00:00:00.000Z). At the bottom is an 'Output' section showing the text 'Server listening at 192.168.0.41:8888'.

