

The Response File Responds Silently

By Nick Decker

The recent Advantage article *Without a Trace – The Silent Installation* revealed the many benefits of a silent installation and how it provides administrators with the ability to install software across the network quickly and automatically. The article also discussed the user-configurable options within the **.properties** file. While that silent installation paradigm worked well for the BBJ® 4.x series of product, the release of BBJ 5.0 introduced many new features, enhancements, and changes – one of which was the new installation. BASIS chose to rewrite the installation from scratch for BBJ 5.0, using the industry-standard InstallShield Universal product rather than InstallAnywhere used with the BBJ 4.x installation. Why? Along with many new capabilities and extensive componentization featured in another Advantage article *New Install Puts You in the Driver's Seat*, this latest installation process provides a more powerful mechanism for silent installations – the response file.

The response file mechanism has several benefits over the older **.properties** mechanism. One of the primary features of the response file is the ability to customize the entire installation process. Since the response file contains the answer to every single question presented during installation, the developer can store such answers as where to install BBJ and whether it is a typical or custom install. Additionally, the developer can choose whether to install the dozens of components in BBJ's heavily componentized custom installation, for high-level components such as the documentation, and lower level components such as standard, extended, or BASIC Web utilities. Pre-programming all of these choices into the response file makes it possible to build a custom installation/response file for any deployment situation or customer.

The response file may seem exceedingly powerful and intimidating with over 50 customizable options available, far more than the five options in the previous **.properties** file. However, InstallShield presents an uncomplicated way to create the response file – simply perform an installation with the special **record** flag. It is similar to recording a macro in other software products; start the recording process, perform the installation, and end up with a file that contains all the necessary information. The newly created response file runs on any supported platform. In other words, it is possible to use the response file generated on a Mac OS X machine to run a silent install on a Microsoft Windows system. Because the response file is a heavily commented ASCII file, it is commonplace to generate the file once and then modify it for use in different deployment scenarios. Simply turn specific options on or off by setting the associated variable to **true** or **false**.

In addition to listing the product components selected for installation, the response file includes a few sections devoted to controlling the BASIS installation and configuration wizards. These same wizards run interactively from BBJ's Admin program and provide functionality such as the ability to register for and install

BASIS Product Licenses. Because configuration is an integral part of installing and using any product successfully, it is possible to create a response file that optionally runs the most commonly used wizards, but in a silent and scripted fashion.

To make it even more flexible, set the response file to launch one or more of the wizards interactively or silently. For example, you may want your customer to register for a BASIS license immediately after installing the product. In this case, after the silent product installation, force the licensing wizard to run interactively. However, if you already registered for and received a BASIS Product License, then force the installation to silently install the BASIS License Manager, set it to run automatically at startup as a Windows Service, install the product license, and start up the BLM.

So how exactly does one create a response file? As mentioned above, it is a matter of performing an installation in GUI mode with an extra command line parameter. Here is the command line under Windows:

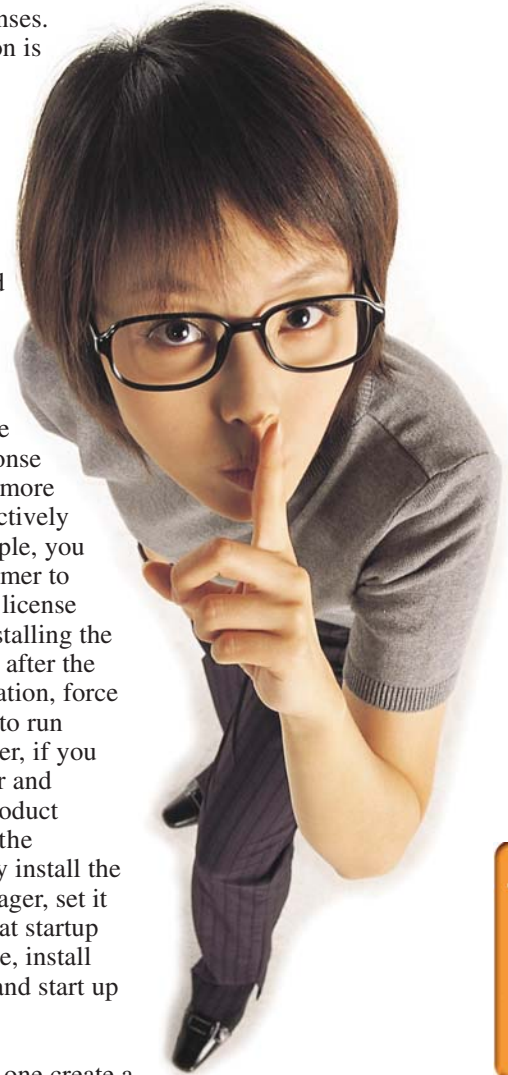
```
windows.exe -options-record C:\response.file
```

In this example, instead of only running the **windows.exe** executable, add the **-options-record** command line parameter and follow it with the name of the response file. The installation continues as normal, except that upon completion, it creates the response file named **C:\response.file**.

Component Selection

Examining the file reveals several sections devoted to the product component selection. For example, the sample in **Figure 1** shows selections devoted to the documentation feature. This is a prime example of how to use a simple true and false value to configure which components to install. The lines at the top of this section begin with the comment character **#** and provide useful information about the feature, including how to turn it on and off.

continued...



Partnership

Language/Interpreter

Database Management

Development Tools

System Administration



Nick Decker
Engineering
Supervisor

```
#####
#
# "Documentation" Feature
#
# The selection state of the "Documentation" feature. Legal values are:
#
# true - Indicates that the feature is selected for installation
# false - Indicates that the feature is not selected for installation
#
# For example, to select "Documentation" for installation, use
#
# -P documentationFeatureBean.active=true
#
#
# -P documentationFeatureBean.active=false
```

Figure 1. Documentation section from a sample response file

Installation and Configuration Wizards

The remaining sections deal with the BASIS Installation and Configuration Wizards and are a bit more involved since they offer more choices than simply turning a component on or off. The response file presents the wizards in three sections; the first section (Figure 2) addresses the wizards that can run interactively or not at all. The comment section indicates that you can configure the wizards to run interactively, such as the License Registration wizard is set to do here. If you do not want a wizard to run, just set the value to **Off**.

```
#####
#
# BASIS Installation and Configuration Wizard options
#
# The following variables set whether or not to run some of the various BASIS
# installation and configuration wizards after the installation of the software.
# Setting a value to 'Interactive' will cause the specified wizard to be run
# interactively. Setting a value to 'Off' will prevent that wizard from being
# run.
#
# Values: Off|Interactive
-V IS_WIZ_LICENSE_REGISTER="Interactive"
-V IS_WIZ_EM="Off"
-V IS_WIZ_FINISH="Off"
-V IS_WIZ_REBOOT="Off"
```

Figure 2. Setting interactive wizards in the sample response file

The second section lists six more wizards that differ from the previous group because they can run silently in addition to interactively, or not at all. This may not seem like a significant difference, but a wizard must have all of the necessary information ahead of time to run silently, since it cannot rely on a user interface popup to query the user. Fortunately, the response file is a perfect place to store the information that is essential to execute the wizards silently. The example in Figure 3 shows how to configure the BLM and start it automatically.

continued...

```
#####
#
# BASIS Installation and Configuration Wizard options
#
# The following variables set whether or not to run some of the various BASIS
# installation and configuration wizards after the installation of the software.
# Setting a value to 'Interactive' will cause the specified wizard to be run
# interactively. Setting a value to 'Off' will prevent that wizard from being
# run. Setting a value to 'Silent' will cause the wizard to run silently,
# provided the appropriate properties are set in the next section. Currently,
# the installer runs all Silent wizards first, then all Interactive. Therefore,
# it's not recommended that you mix and match Silent and Interactive wizards,
# unless the ordering makes sense. In other words, do not run the
# IS_WIZ_BBJ_START_STOP silently and the IS_WIZ_BBJ_CFG_STARTUP interactively,
# as this will cause BBJServices to be started before the wizard is shown to
# configure its startup.
#
# Values: Off|Interactive|Silent
-V IS_WIZ_LICENSE_INSTALL="Off"
-V IS_WIZ_BLM_CFG_STARTUP="Silent"
-V IS_WIZ_BLM_START_STOP="Silent"
-V IS_WIZ_BBJ_BRAND="Off"
-V IS_WIZ_BBJ_CFG_STARTUP="Off"
-V IS_WIZ_BBJ_START_STOP="Off"
```

Figure 3. Setting silent wizards in the sample response file

In order for the two BLM wizards shown in **Figure 3** to run **silent** properly, they require additional information presented in last wizard section shown in **Figure 4**.

```
#####  
#  
# BASIS Installation and Configuration Wizard options  
#  
# The following variables define properties to be used in conjunction with one  
# of the above wizards when that wizard is set to 'Silent.'  
#  
  
# Value: hostname of the machine running the BLM  
-V IS_WIZ_BBJ_BRAND_SERVERNAME=""  
  
# Values: Service|Login|Manual|InitTab  
-V IS_WIZ_BBJ_CFG_STARTUP_TYPE=""  
  
# Values: Automatic|Manual|Disabled  
-V IS_WIZ_BBJ_CFG_STARTUP_SERVICESTARTUPTYPE=""  
  
# Value: Windows User Account that BBJ Services will use to run  
-V IS_WIZ_BBJ_CFG_STARTUP_USERACCOUNT=""  
  
# Value: Windows User Password that BBJ Services will use to run  
-V IS_WIZ_BBJ_CFG_STARTUP_PASSWORD=""  
  
# Values: True|False  
-V IS_WIZ_BBJ_START_STOP_STARTUP=""  
  
# Value: full path to the license file to install  
-V IS_WIZ_LICENSE_INSTALL_LICENSEFILE=""  
  
# Values: Service|Login|Manual|InitTab  
-V IS_WIZ_BLM_CFG_STARTUP_TYPE="Service"  
  
# Values: Automatic|Manual|Disabled  
-V IS_WIZ_BLM_CFG_STARTUP_SERVICESTARTUPTYPE="Automatic"  
  
# Values: True|False  
-V IS_WIZ_BLM_START_STOP_STARTUP="True"
```

Figure 4. Defining properties for the silent installation

The last three variables in **Figure 4** supply the information required for the BLM wizards – how to configure the BLM and whether to start it up at the end of the installation. The **Service** setting for the variable **IS_WIZ_BLM_CFG_STARTUP_TYPE** installs the BLM as a Windows service. Options for this variable include:

- Service** – Run as a Windows service on NT-based operating systems
- Login** – Run when the user logs in to the Windows machine
- Manual** – Require the user to start the BLM manually (does not start automatically)
- InitTab** – Run via the **inittab** or **init.d** on UNIX and Mac systems

The second variable **IS_WIZ_BLM_CFG_STARTUP_SERVICESTARTUPTYPE** configures the BLM Windows service to start up automatically. When the BLM service runs as a Windows service, you may configure it to run automatically, manually, or set it to a disabled state. These states are available with every Windows service and are modifiable after the installation via the Admin program or the Windows Service Control Manager.

The last variable **IS_WIZ_BLM_START_STOP_STARTUP** set to **True** starts up the BLM before the installation completes. With the correct settings and necessary configuration information, other wizards can run silently just as the BLM wizards do. By following these same steps you can configure BBJ Services, install a license, and brand BBJ to a particular machine so that it always requests its license from a BLM on that machine.

After creating a customized response file, the next step is to test it. Running the install silently with a response file is similar to the process of creating a response file – run the installation and provide a couple of command line parameters. For example:

```
windows.exe -options C:\response.file -silent
```

continued...

This time, instead of providing the `-options-record` parameter, use `-options` to specify which response file to use during the install instead of creating a file. The last parameter, `-silent`, instructs the installation to run silently without any user interface. After executing the command, the machine appears busy for a short while, depending on the number of components configured for installation.

A common question for a silent install is “Once it’s installed, how can I tell what happened?” Since the user never has an opportunity for feedback, how can the developer tell what occurred during a silent installation? First, check the target directory to ensure each of the appropriate components installed correctly. A successful Windows, Mac, or Linux silent installation will also result in usable shortcuts for the installed components. Sometimes, however, it is desirable to find out much more detailed information about what happened during the installation process. The installation program automatically creates several log files in the `<TARGET_DIRECTORY>/_inst` directory that detail the installation and uninstallation process. These log files contain information about the target system such as the operating system name and version, installation source and target paths, the location of Java on the machine, the locale of the system, and much more. In addition, these log files detail each component’s installation selection, the configuration wizards selected to run, and if the wizard ran successfully.

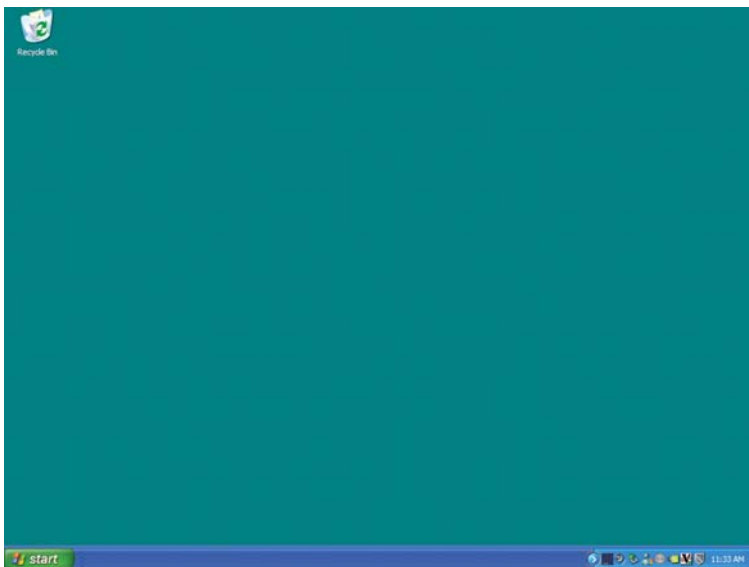


Figure 5. A silent Windows install in action



For more information about silent installation, read *Without a Trace – The Silent Installation* at www.basis.com/advantage/mag-v8n1/silent.html.

BASIS introduced the new install with BBJ 5.0, read *New Install Puts You in the Driver’s Seat* at www.basis.com/advantage/mag-v9n1/install.html.

Summary

BBj 5.x brought about a host of new features and enhancements, one of which was an improved and extensive silent installation based on a powerful new response file format. While BBJ 4.0’s installation touched on the capabilities of a silent install, the BBJ 5.x installation fully exploits all of the possibilities of a silent installation. Developers can now completely control nearly every aspect of the installation - from the target directory, to which components to install, to which configuration wizards to run. It is a shame something of such beauty and power will never be seen by the typical end user (see **Figure 5**). But then, that’s the point! 