

# Revolutionize your application and product support

## An overview of the IBM Application Runtime Expert for i

Mark Schroeder

April 12, 2011

The IBM Application Runtime Expert (ARE) is an exciting new product created exclusively for IBM i. ARE has the potential to make you re-think your entire approach to system management, application problem determination, product support, and testing.

### Background

The 21st century has seen a dramatic change in both the software and hardware industries. Hardware virtualization has allowed customers to take a single system and carve it up into many different partitions. Software is increasingly being built on top of complex foundations or frameworks. Many times, this complex software runs in its own virtual machine (such as a JVM), which itself may be running on virtualized hardware resources. The bottom line is that systems, and the software running on them, are much more sophisticated and complex than they used to be.

So within this new landscape, what happens when something does not work correctly? Perhaps it was working correctly and then stopped working. Maybe it does not work on one partition but works fine on five others. Maybe the application still runs, but the performance is terrible. Why isn't it working? What has changed? What is difference between the working and non-working systems? With the increase in application and system complexity, the answer to this question is often not simple or quick.

The IBM Application Runtime Expert for i (ARE) is a new product, built specifically for the IBM i, that can help you answer the critical "what is wrong with the application" question. ARE allows you to create an automated, repeatable process for determining what has changed on a system, or the applications running on that system. ARE also provides the capability to quickly and easily compare attributes, applications, systems, and environments, between systems or partitions. This is enormously helpful when diagnosing issues where something works on one system but not another. The best part is that leveraging ARE to do all of this is quick and easy.

### Product overview

Before we get started, it is important to note that to have access to all of the features discussed in this article, you must have IBM Application Runtime Expert for i PTF 5733ARE SI42401, or newer,

installed on your system. PTF SI42401 upgrades the web user interface for Application Runtime Expert to version 1.5.1.

## A simple example

Let's first take a look at a simple example to help you understand what ARE can do for you. Let's say you have an application that consists of 1,000 files. These could be files in IFS, the Library system, or a mixture of the two. If the authority on just one of these files or directories gets modified, it may cause your application to stop working. Using ARE, you can build a template of the working application. When the application stops working, ARE can use the template you built to compare the good, working state of the application against the non-working application. So in a matter of minutes, ARE can find and fix (yes, it can correct issues for you!) the authority issue that is causing your application to fail.

Not convinced? Perhaps authority issues are not very interesting to you? OK, let's say you have a set of configuration files that contain values your application uses to control its behavior. If one attribute in one of these files gets modified, you could spend hours trying to look through all these files to find what is different. ARE can accomplish that same task in minutes. We could repeat this example over and over for all of the different attributes that ARE can verify. The point is that the power of ARE is not just its ability to compare and verify, but that you can build an entire application profile that can be used to verify your *entire* application *every* time.

## Deployment template editor

The ARE web user interface consists of two web applications: the deployment template editor and the console. The template editor allows you to create and modify the contents of deployment templates. The console provides a way to use a template to verify any IBM i system on your network. The web applications are accessible via this URL:

```
http://hostname:12401/are/index.html
```

Before we dive in and start talking about the deployment template editor, let's start with a little terminology. What exactly is a deployment template? Let's actually simplify that even a bit more by first taking a look at the term 'deployment' to understand exactly what that means. A deployment is a collection of attributes about both software and the environment where the software runs. Common examples of a deployment are:

- Any IBM i product
- Any application on the IBM i
  - An ISV or internally written application
  - A WebSphere Application Server
  - An integrated web application server
  - An RPG application
  - A directory tree
- A custom selection of software, system, and environment information

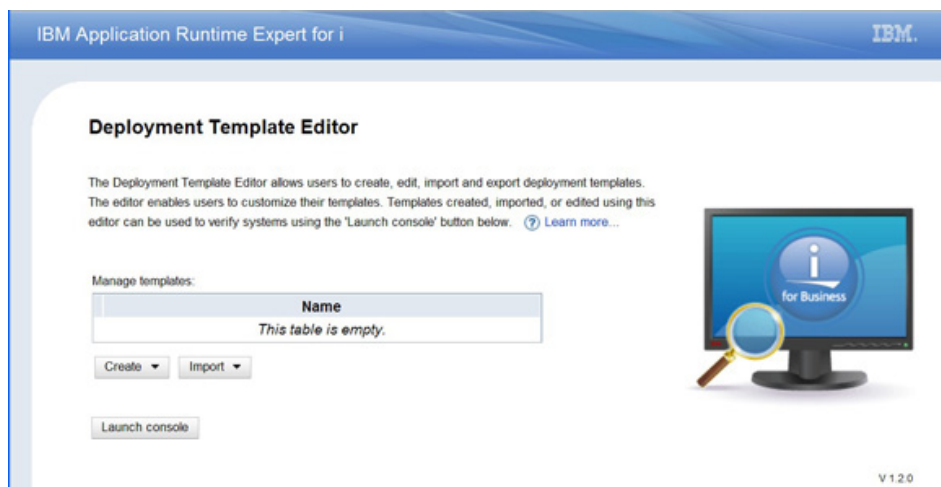
Put simply, a deployment is the application, system, and environment information you wish to verify.

A template is the collection of attributes for the deployment. The attributes can be specific to your *application*, such as directory, file, and library information (size, existence, create and modify dates, symbolic links, all authority related attributes); *system information*, such as user IDs, port information, system values, PTFs; or *environment information*, such as network integrity, contents of configuration files, dependent products, or environment variables. Once you have built a template, it can be used to verify the condition of your deployment on the same, or another, IBM i system. All of the information necessary to verify the system is contained within the template. In the next section, we'll take a look at the web user interface that is used to build and manage templates.

## Creating a new template

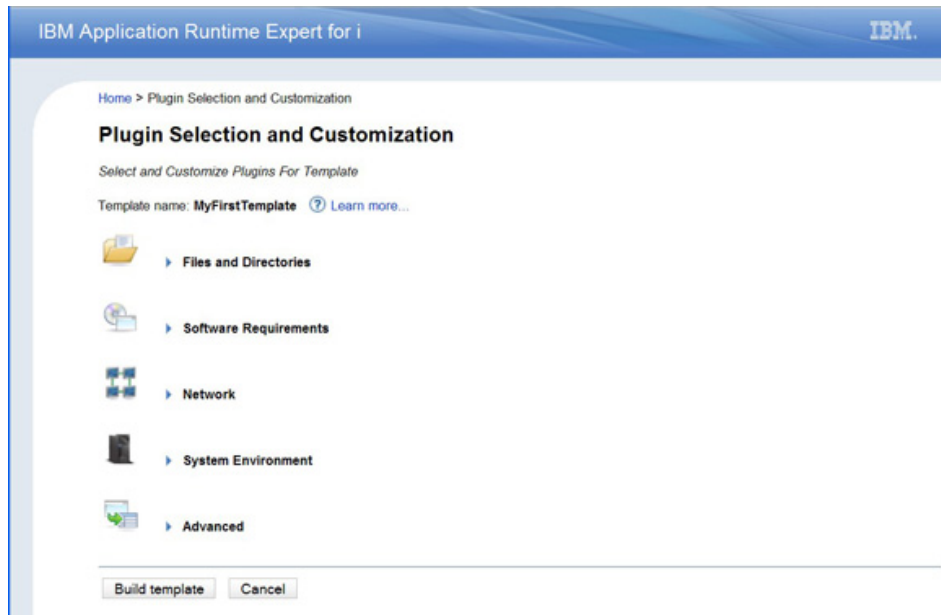
When you log into the ARE web user interface, you are directed to the deployment template editor home page, as shown below in Figure 1.

**Figure 1.**The deployment template editor home page



There are, upon first usage of the ARE web user interface, no templates to display or work with. Therefore, the table listing templates on the system is empty. To create a new template, click the Create button. All you need to provide is a name for your template. Once you have that, click the Create button and you are ready to go. Clicking the Create button takes you to the Plugin Selection and Customization page (see Figure 2).

## Figure 2. Plugin Selection and Customization page



The Plugin Selection and Customization page presents you with five different categories of plugins. A plugin is something that performs verification for a specific type of application or system attribute. For example, there is a File and Directory Authorities plugin that can verify authorities for files, directories, and library objects on an IBM i system. Another example is the user IDs plugin for verifying attributes of IBM i user IDs, and so on. To view the available plugins in each category, click on the category icon or name.

One great feature of the GUI is that if you are unsure about something, you can always get more information about it by clicking on the [Learn more](#) link. Learn more links are available throughout the GUI to provide context specific help wherever you are at.

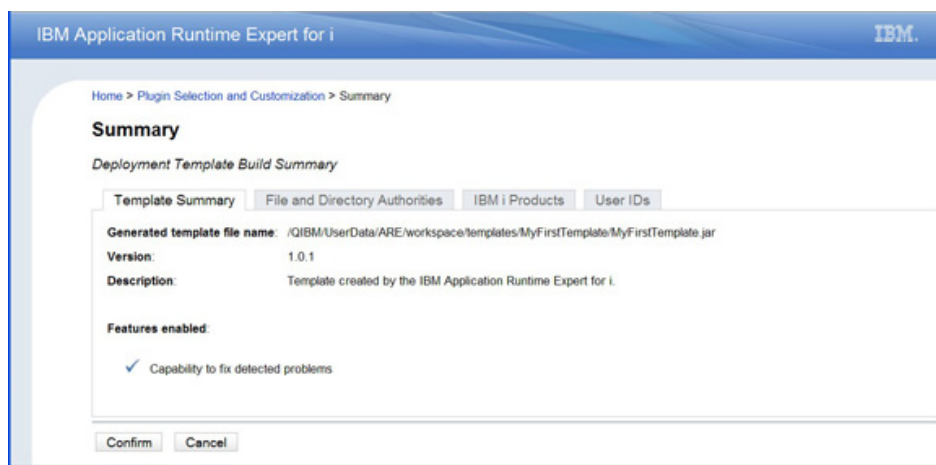
The template editor offers plugins that can verify a wide range of application attributes, as well as plugins that focus on IBM i system level attributes (including networking). If your application has unique attributes that need to be verified, or perhaps there is something ARE has just not yet provided, that's no problem. You can easily write your own custom plugins, which can be seamlessly integrated directly into any template.

Building your template is as simple as selecting the plugins you care about and then selecting the attributes that pertain to the application that you want to verify. If you want to verify object authorities, you would select the File and Directory Authorities plugin. From within that plugin, you can browse the entire file system to select the files and libraries you want to verify. If you want to verify user IDs, you can select which user IDs you want to verify, and even specify what attributes of the user ID you would like to verify. This same concept holds true for the rest of the plugins: you get to select exactly what *you* want to verify. And therein lies the power and potential of ARE; practically every application has its own, unique set of files, system values, user profiles, products, etc that it depends upon, so ARE allows you to specify what is important to verify for your application. You get the flexibility you need to verifying your application is correct. And the

best part is that all of this selection and customization is done via the web user interface, which makes building templates fast and easy.

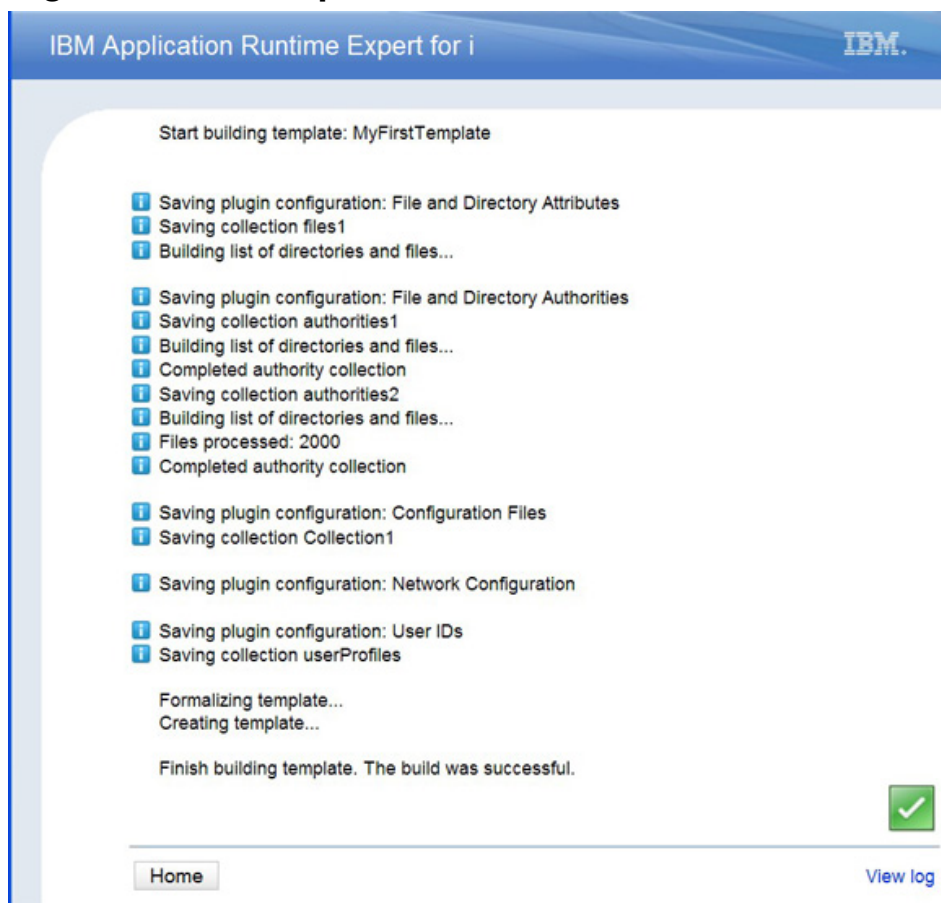
Once you have selected what you want to verify, all that's left to do is build the template. Don't worry if you haven't selected everything you want, or are unsure about a few of the items that you did select. The template editor assumes you will need to edit your templates after they have been built, so the capability to edit an existing template is already there for you. To build your template, click on the Build template button at the bottom of the Plugin Selection and Customization page.

### Figure 3. Template build summary page



The summary page has tabs for each plugin you selected, which summarize the selections and customizations you made. This is a great opportunity to review your selections; if anything is wrong or missing, click the Cancel button to return to the Plugin Selection and Customization page. Once you have confirmed your choices, click the Confirm button. This initiates the build of the template, which leads to the obvious question: what is the GUI doing to actually build the template? During template build, the GUI is actually mining information about the attributes/items you selected from this system. For example, if you select files to verify authority attributes, the GUI mines the owner, primary group, etc attributes for each file you selected. All of this information is stored in a series of XML files, which describe the items you selected and all of the attribute information mined by the GUI. Once the XML files are generated, they are packaged, along with other configuration information needed by ARE, into a JAR file. This JAR file is your deployment template.

Depending on several factors (system speed, available memory, number of items selected, etc) your template often takes less than a minute to build, but could take several minutes. One very nice feature of the template build process is that the entire process is displayed for you in real time! See Figure 4 for an example of what the build status page looks like.

**Figure 4. The template build status**

Once the template has been built, clicking the Home button returns you to the deployment template editor home page. And that's it! You have now successfully built your first template. The template now appears in the Manage templates table. Also, there is also new button available: Actions. The Actions button lists all of the actions you can perform on the selected template, such as copying, editing, exporting, and so on.

## Updating existing templates

It is expected that you will need to update templates after you build them, so updating an existing template is a simple process. From the Manage templates table, select the *Edit* action. This returns you to the Plugin Selection and Customization page, with all of the attributes you previously selected available for you to change, remove, modify, or add. Just make your updates and re-build the template. The build of an existing template is much quicker too, since only modified or added attributes are updated during the build process. So overall, updating an existing template is quite simple, just as it should be.

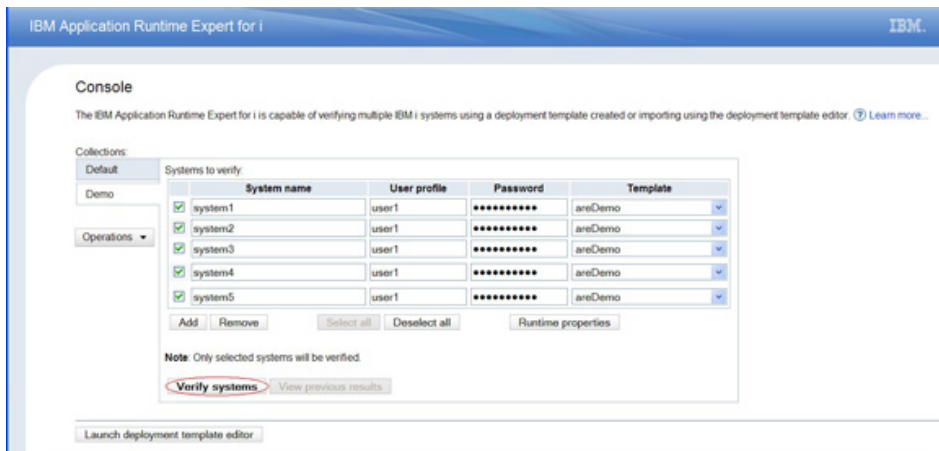
## Console

Once you have built a template, you will want to use it to verify your application on a system. There are two options for doing this: use the GUI Console or a Qshell script. We'll only talk about the Console here but more information about the script interfaces to ARE can be found on the ARE product web site listed the [Resources](#) section.

The ARE console, or just console, is the web user interface where templates can be used to verify an application on either the local system or any IBM i system in your network. One of the primary benefits of using the console is that no templates need to be transferred to the target systems; the console takes care of doing that for you. All you need to do is select a template to use, provide the target system name or IP address, user name, and password. The console uses the information you provide to verify the target system(s). The results of the verification are retrieved from the target system so that you can review the results directly from your web browser. The console can verify multiple partitions or systems simultaneously, making it very easy to verify applications across your entire enterprise environment from a single place.

The main console page is reached by clicking the Launch console button at the bottom of the Deployment Template Editor page. The main console page is where you to specify information about the systems to verify, as shown below in Figure 5.

**Figure 5. Main console page**

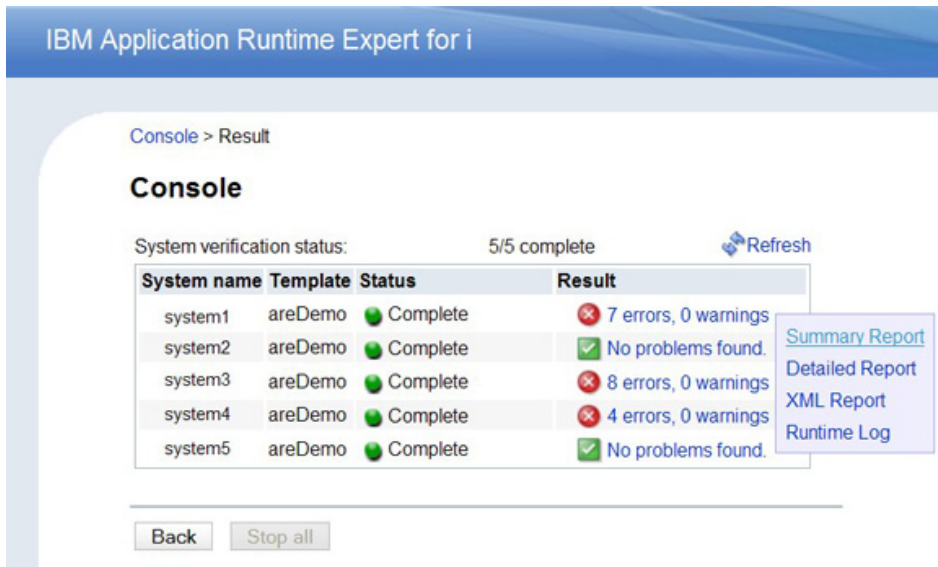


Each row in the table represents one partition or system to verify. You can specify as many as you want, and different templates can be specified for different systems. You start the verification process by clicking the Verify systems button. This takes you to the System verification status page, where you can watch the verification progress of each system. The system verification status automatically refreshes, but you can manually refresh it by clicking the Refresh link in the upper right corner of the verification status table.

Once a system verification is done, a Complete icon is shown in the status column for that system, and a brief summary of the verification results are shown in the Result column. The Result column is much more than just a summary of the verification results; the result column text itself is a link, as shown below in Figure 6. This link provides access to the verification reports generated during the verification of the target system. All three reports—summary, detailed, and XML—can be viewed via the link in the result column. There's also a link to the runtime log. The runtime log contains events written to the console's log during the course of the target system's verification.



**Figure 6. Links to verification reports**



Of the three reports, the most interesting one to view using the console is the summary report. The summary report is a customized, web browser friendly report that contains a summary of all problems detected during the verification of a system. Each row in the summary table contains the results for a specific plugin, such as the Authority Verifier or System Value Verifier plugins. The icon directly before the plugin name indicates the highest severity problem that was found by that plugin. The other items in each row indicate the number of attributes verified by the plugin and the number of problems found at each severity level (error, warning, and info). Details about the problems each plugin found can be viewed by clicking on either the twisty icon in front of the plugin name or the plugin name itself. This expands the summarized view of the plugin and displays detailed information about each problem found by the plugin. If a plugin did not find any problems, then there will be no detailed information available for it.

### Fixing detected problems

The final item in each row, the Fix actions column, indicates how many of the detected problems can be fixed directly from the console web user interface. A fix action is an action that can be taken, in this case directly from the console, to fix a problem detected during verification. This is a very convenient feature that allows you to solve certain problems without ever having to go and log into the target system. It is important to note that only certain types of detected problems, such as authority issues, can be fixed directly from the console. If, say, a file is missing, there's no way to "automatically" fix that from the console.

Figure 7 shows an example of a summary report that has two plugins with fix actions available: Authority Verifier and User Profile Verifier.



## Figure 7. Summary report with fix actions

Console > Result > Summary Report

### Summary Report

System name: system1  
 Remote core version: 1.0.3  
 Template: areDemo  
 User ID: USER1

Summary details [Learn more...](#)

Plugin name	Attributes verified	Error	Warning	Info	Fix actions
File Attribute Verifier	8068	0	0	0	None
Authority Verifier	8074	3	0	0	<a href="#">Fix actions(3)</a>
<b>Description</b> User USER1 has no data authority to object /home/user1 User USER1 has no object authority to /home/user1 Owner of /home/user1 is not what is expected Expected: USER1 Actual: QSYS					
Configuration Attribute Verifier	110	0	0	0	None
Network Verifier	0	0	0	0	None
Network Configuration Checker	0	0	0	0	None
Product Verifier	5	0	0	0	None
User Profile Verifier	54	4	0	0	<a href="#">Fix actions(3)</a>
System Value Verifier	5	0	0	0	None

When you click on the Fix actions link, you are taken to a page that summarizes all of the problems detected by that plugin that can be fixed directly from the console. Once you have selected the problems to fix, simply click the Fix button and the console begins the process of fixing the selected problems on the target system. Once the problems are fixed, a second (results) table is displayed that shows the fix results. Note that in the results table, the first column (Result) shows the result of the console trying to fix the problem, and the second column (Description) contains a description of the original problem, not a description of the fix result.

## Something to try for free?

Over the past year we have found that there are a number of common things that many IBM i users seem to be concerned with when it comes to ensuring their applications and systems are working correctly. To help with this, IBM has shipped a few basic ARE templates that can be used without requiring you to have the ARE product on your system. We currently have four templates available, all of which can be run on a local IBM i system from Qshell.

- **Network:** The Network Configuration plugin verifies a variety of TCP/IP configuration settings and network characteristics. It checks for common configuration issues, much like reviewing the information available via the CFGTCP CL command. This template also performs basic network operations, such as looking up localhost and checking if all configured DNS servers can be reached, in an attempt to verify that the system's network configuration is such that applications that require frequent access to the network, such as Web applications, can do so in a fast, reliable, and repeatable manner.
- **Host servers:** Many applications today require that all of the Host Servers are active on the system. This template ensures that all the different host servers are up and running.
- **Basic integrated web application server:** To use this template, simply specify the name of an existing IAS server on this system, and the template verifies that all of the IBM specific settings and values are set as expected for the IAS server.
- **Pre-checker:** For many customers, this may be the most interesting template we provide. This template is for situations such as if you have a Java based application that seems to be

failing immediately. Or perhaps you are trying to start a Web server (WebSphere Application Server, IAS server, even the ARE server) and the jobs seem to just fail to even start. In these situations, it is possible that Java, Qshell, or PASE may not be working correctly on your system. This is where the pre-checker may be able to help.

The pre-checker is a C-based program that verifies the basic state and condition of the Java™, PASE, and Qshell runtime environments. This template can help diagnose some common issues than can occur, possibly helping you get these environments corrected without spending hours of time doing problem analysis.

These templates are currently available on IBM i 6.1 and 7.1, but they do require the most recent HTTP group PTF (SF99115 on IBM i 6.1, or SF99368 on IBM i 7.1). Once you have the group PTF installed, you can use any of these templates by running the following script from Qshell:

```
/QIBM/ProdData/OS/OSGi/templates/bin/areVerify.sh
```

```
Usage: areVerify.sh <parameter>
```

```
Valid parameter:
```

```
-network           : Verify network configuration and status  
-iasserver server_name : Verify the 'server_name' IAS server  
-precheck         : Verify software products that are necessary to use  
                   the IBM Application Runtime Expert for i  
-hostservers      : Verify the host servers are active
```

## Summary

The IBM Application Runtime Expert for i provides you with the tools to rapidly understand your applications and the environments they run in, as well as to verify these applications in a repeatable manner that has the potential to save you incredible amounts of time when servicing your applications and the systems they run on.

See the [Resources](#) section for links to much more information about the IBM Application Runtime Expert for i product.

© Copyright IBM Corporation 2011

([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))

[Trademarks](#)

([www.ibm.com/developerworks/ibm/trademarks/](http://www.ibm.com/developerworks/ibm/trademarks/))