IBM Application Runtime Expert for i

Understanding ARE Reports

Background

The IBM Application Runtime Expert for i core, or ARE core, is the ARE component that verifies a system using templates built using the deployment template editor. Once verification of a system has completed, three different reports are generated. The format and content of these reports varies. In this document, we will discuss in detail the three reports generated by the ARE core once verification is complete.

There are two ways to use a template to verify a system:

1. Use the console Web user interface
2. Use a script that can be run from QShell

Both of these verification options will result in the same three reports being generated. When verifying a system using the console Web user interface, the user interface makes viewing the different reports very easy. If you verify a system using the QShell script, three output files are generated; the names of the output files are based on input provided to the runARE.sh script. It is important to understand which output file corresponds to which report, so let's look at a quick example. Let's say you ran a verification using the following command:

```
/QIBM/ProdData/OS/OSGi/healthcheck/bin/runARE.sh -template
myTemplate1.jar -outFile report.out
```

Based on the value specified for the -outFile parameter, the following set of output files are generated:

- report.out – This is the detailed report
- report.out.summary.txt – This is the summary report
- report.out.xml – This is the XML report

It is important to note that if only a file name is provided for the -outFile parameter, all output files will be placed in the current working directory. If, however, a fully qualified path and file name is provided, then all output files will be placed in the location specified by that fully qualified path.

Details about each of these reports are discussed in the subsequent sections of this document, but before discussing the details of the reports, we first need to discuss the concept of problem severity, and how a problem's severity is reflected in the reports ARE core generates.

Problem Severity

Each check that is done during verification has a severity level associated with it. If a problem is detected, the problem is reported using the severity level associated with the check that detected the problem. Problems can be reported at three different severity levels:

- Error (highest severity)
- Warning
- Info (lowest severity)

In the summary and detailed reports, the problem severity is reported as the first word in the problem message. For example:

Error! This is a problem reported at the error severity level
Warning: This is a problem reported at the warning severity level
Info: This is a problem reported at the info severity level

Because problems are prefixed with their problem severity, it is very easy to differentiate between regular (status) messages and problem messages.

Summary Report

The summary report is a text report that summarizes only the problems that were reported during verification of a system. This report provides a quick snapshot of the most critical information (the problems found) generated during verification of a system.

Due to the way in which reporting is structured, it is often the case that reported problems can only be fully understood if some context of the problem is provided. To account for this, each problem in the summary report also contains a select number of status messages which were reported just prior to the problem being reported. To help illustrate how these contextual messages may look, let’s take a look at an example. The example below is what the authority verification portion of the summary report would contain if a single authority problem was reported:

```
Running plugin Authority Verifier
   > Processing XML rule file (xml/Auth_Collection1.xml) for object authority
      o Checking authority for /QIBM/ProdData/OS/OSGi/apps
         - OK: Authorization list set to *NONE for /QIBM/ProdData/OS/OSGi/apps
ERROR! User *PUBLIC data authorities to object /QIBM/ProdData/OS/OSGi/apps is
not what we expect
Details: Expected: *RX Actual: *EXCLUDE
   > Finished processing object authority
   > Total number of object authority items checked: 1286
```

In the above example, the actual problem reported is in blue; as you can see the reported problem is not the only item included in the summary report. Also included are a few status messages that were reported just prior to the problem, as well as a couple of messages summarizing the number of items checked during the authority verification.

It is important to understand that because the summary report only contains information about problems that were found, it is possible that this report may not contain all of the interesting information generated by the verification. For example, the PTF plugin reports the group PTF level for all group PTFs installed on the system. This information is not reported as a problem (because it is not), and therefore it is not included in the summary report. This information would, however, be included in the detailed report.

Detailed Report

The detailed report is a text report that contains every status and problem message reported during verification. This means the report is a *complete* record of everything that was checked during verification and the result of each check, even if the check did not detect a problem. Because this report contains such detailed information, it can grow quite large if many items are verified. This can make viewing the report, and finding specific pieces of information in it, a challenge. There are, however, a few tricks that can greatly reduce the amount of time it takes to search through the report to find information you are interested in. A few basic tips are discussed bellowing using a question and answer (Q & A) format.

Q: How do I quickly find reported problems in the detailed report?
A: The easiest way to find reported problems is to search for the severity level text that will precede any reported problem. As discussed in the Problem Severity section, depending on what severity a problem is reported at, one of the following three strings of text will precede the reported problem:

ERROR!
Warning:
Info:

Q: How can I find the place in the report where the results for a specific type of verification, such as verifying PTF levels or system values, are located?
A: Application and system attributes (authority, PTF levels, system values, user IDs, etc) are verified by plugins. Each plugin is responsible for verifying a different attribute; so PTF levels are verified by a PTF plugin, system values are verified by a System Values plugin, and so on. Each plugin has an entry in the detailed report that marks the beginning of verification by that plugin. So the best way to find results for a specific type of verification is to find the beginning of verification of each plugin. This can be done by searching the detailed report for the following text:

Running plugin

Each time this text is found, it marks the beginning of verification for a different plugin.

XML Report

The XML report is an XML formatted report that contains every status and problem message reported during verification. This means the report is a *complete* record of everything that was checked during verification and the result of each check, even if the check did not detect a problem. In this regard, the XML report is exactly like the detailed report, except in an XML format instead of plain text. However, there is additional information in the XML report that is not included in any other report: information about how to fix detected problems.

The details about how ARE fixes detected problems are not discussed in this document. More information about that can be found in the ‘How To Automatically Fix Detected Problems’ document on the ARE product Web site:

http://www.ibm.com/systems/power/software/i/are/documentation.html

Because the XML report contains information about how to fix detected problems, typically the primary usage of the XML report is by the ARE core; it uses the report as a guide for automatically fixing detected problems. If you are using the ARE console to fix detected problems, the XML report will be used “under the covers” to guide the fixing of the problems. If, however, you are using the areFix.sh QShell script to fix detected problems, then you will need to provide the XML report as input to the areFix.sh script.

For more information about the areFix.sh script, as well as other ARE QShell scripts, see the ‘Script Interfaces to ARE’ document on the ARE product Web site:

http://www.ibm.com/systems/power/software/i/are/documentation.html