Contents

- Listing Installed Fixes 1
- Using dbintegritycheck to Check the Database 2

Important! The functionality described in this supplement requires MWS_8.2_SP1_Fix9 or later. You can obtain MWS_8.2_SP1_Fix9 from Software AG Update Manager or the Empower Product Support Web site.

Listing Installed Fixes

You can determine the fixes that have been applied to My webMethods Server using the following methods:

- Review the _full_.log file.
- Execute the envcapture command line tool.
- Execute the loganalyzer command to view fixes associated components that have errors.

Reviewing the _full_.log File

My webMethods Server determines the installed fixes at server startup and logs the list of installed fixes to the _full_.log file. My webMethods Server log files reside in the following directory:

My webMethods Server_directory\server*server_instance*\log

For more information about this log, see *Administering My webMethods Server*.

Because determining the list of fixes might take a few minutes, you might want to disable this action at server startup. To prevent My webMethods Server from listing installed fixes at startup, edit the `startupDiagnostics` section of the phaseProvider.xml configuration file as shown below.

```
<Phase class="com.webmethods.portal.system.init.impl.DefaultPhase"
                enabled="true" name="startupDiagnostics">
       <PhaseInfo class="com.webmethods.portal.system.impl.UpdatesInfo"
                enabled="false" name="updatesInfo"/>
    .
    .
    .
</Phase>
```

The phaseProvider.xml configuration file is located in the database or under *My webMethods Server_directory*\server*server instance*\config. For more information about the phase provider file, see *Administering My webMethods Server*.

Using envcapture to List Installed Fixes

The envcapture tool provides environment information for My webMethods Server, such as, information about log files, information about configuration files, and details about the local operating system and Java virtual machine. The envcapture tool now also provides a list of fixes that have been applied to My webMethods Server. For more information about envcapture, see *Diagnosing My webMethods Server*.

Using loganalyzer to List Installed Fixes Associated with Components with Errors

The loganalyzer tool analyzes reported issues in log files, for example, the errors.log file. If you specify that you want loganalyzer to provide component-oriented output that displays the components that cause the events in the log, loganalyzer also lists fixes, if any, that are applied to the components.

Using dbintegritycheck to Check the Database

My webMethods Server supports a new database integrity check tool named dbintegritycheck. The dbintegritycheck tool reviews the dynamic business objects (DBOs) deployed to My webMethods Server to search for potential errors that might have occurred if a DBO was installed or upgraded incorrectly.

My webMethods Server automatically executes dbintegritycheck at the following times:

- After a My webMethods Server FIX installation
- When a My webMethods Server FIX is rolled back
- After an over-install upgrade of My webMethods Server
- During My webMethods Server startup

You can also run dbintegritycheck as a standalone command line tool on Windows and UNIX environments.

To check the integrity of a deployed DBO, dbintegritycheck compares the VARCHAR column data type of the deployed DBO with the metadata for the DBO. If a difference is encountered, dbintegritycheck reports the discrepancy and provides the table name, column name, and data type associated with the discrepancy.

Syntax

```
Windows: dbintegritycheck.bat
UNIX:    ./dbintegritycheck

[{-h|--help}]
[{-l|--reportLevel} [quiet|full]]
[{o|--output} directory]
[{-s|--server} name]
[{-x|--xml}]
```

Options

Option	Description
`{-h|--help}`	Displays information about the tool and the available optional arguments.
`{-l|--reportLevel} [quiet|full]`	Indicates the level of detail you want in the report. Specify one of the following: ■ `quiet` indicates that you want dbintegritycheck to list only the tables, columns, and data types that did not match, that is report only the discrepancies. This is the default. ■ `full` indicates that you want dbintegritycheck to list all tables, columns, and data types and a message stating whether the information matched, indicating it is valid, or did not match, indicating a discrepancy.
`{-o|--output}` *`directory`*	Stores the resulting output to a directory you specify. Supply the directory path where you want the tool to write the output. By default, if you do not use the `{-o|--output}` option, dbintegritycheck stores the results in the same directory in which you started the command.
`{-s|--server}` *`name`*	Executes dbintegritycheck on another My webMethods Server instance that is available on the local system. Supply the name of the server instance for *`name`*. By default, if you do not use the `{-s|--server}` option to specify an alternative server instance, the tool executes on the default instance.

Option	Description
`{-x|--xml}`	Generates the results in XML files rather than displaying the results in the console window. The tool creates the following two XML files: ■ Standard log4j XML report You can execute loganalyzer against this XML file. ■ Structured XML report You can use this XML file with the envdiff tool. The dbintegritycheck tool creates the XML files in the same directory in which you started the command.

Usage Notes

- To use the dbintegritycheck as a standalone command line tool, at the command line prompt use the `cd` command to move to the following directory where the dbintegritycheck resides:

  My webMethods Server_directory`\tools\diagnostics\bin`

  Once positioned at the directory, at the command line prompt enter the dbintegritycheck command using the syntax described above.

- When dbintegritycheck exits, it provides one of the following exit code values:

Exit Code Value	Meaning
0	The dbintegritycheck tool found no errors in the database.
1	The dbintegritycheck tool found discrepancies in the database.
2	The dbintegritycheck tool encountered errors unrelated to the database or cannot be executed for some reason.

- You can analyze the results of dbintegritycheck using the loganalyzer tool. To do so, execute dbintegritycheck using the `{-x|--xml}` option so that it produces XML files containing its results. Then you can execute loganalyzer specifying the location of the XML file that is in log4j format.

  The following shows an example loganalyzer command line that uses the default action, `frequency`, to capture log information by the frequency of appearance. This creates a frequency report from log4j XML file produced by dbintegritycheck. The sample loganalyzer command line also includes the `--severity ERROR` option to narrow down the results so that the frequency report only includes items with severity level ERROR:

```
loganalyzer.bat --severity ERROR
C:\softwareag\MWS\tools\diagnostics\bin\db_integrity_report.xml
```

For more information about loganalyzer, see *Diagnosing My webMethods Server*.

- You can compare the results of two different executions of dbintegritycheck using the envdiff tool. To do so, execute dbintegritycheck using the `{-x|--xml}` option so that it produces XML files containing its results. You can use envdiff against the structured XML file that dbintegritycheck produces.

  For example, after a clean installation of My webMethods Server, you might execute the following dbintegritycheck command to create a full report so that you have a baseline structured XML file for reference.

  ```
  dbintegritycheck.bat -l full -x
  ```

  You save a copy of the baseline file by renaming the resulting structured XML file to baseline_db_integrity_check.xml. Then at a future time, if you encounter a problem, you can execute the same dbintegritycheck command again to create a full report that contains the database errors. You can then use the envdiff tool to compare the differences in the two reports. For example, you might use the following command:

  ```
  envdiff baseline_db_integrity_check.xml structured_db_integrity_check.xml
  ```

  For more information about envdiff, see *Diagnosing My webMethods Server*.

- The My webMethods Server automatically executes dbintegritycheck at startup and logs the results to the _full_.log file. My webMethods Server log files reside in the following directory:

  My webMethods Server_directory\server*server_instance*\log

  Although it is recommended that you keep this action enabled, you can prevent My webMethods Server from executing dbintegritycheck at startup. To do so, edit the `startupDiagnostics` section of the phaseProvider.xml configuration file as shown below.

  ```
  <Phase class ="com.webmethods.portal.system.init.impl.DefaultPhase"
                enabled="true" name="startupDiagnostics">
       <PhaseInfo
            class="com.webmethods.portal.system.impl.DbIntegrityCheckLauncher"
            enabled="false" name="dbIntegrityCheck"/>     .
       .
       .
  </Phase>
  ```

  The phaseProvider.xml configuration file is located in the database or under *My webMethods Server_directory*\server*server instance*\config. For more information about the phase provider file, see *Administering My webMethods Server*.

Example

- To execute dbintegritycheck on Windows against the server instance named myServer and use the default reporting level that only shows discrepancies, specify the following command:

  ```
  dbintegritycheck.bat -s myServer
  ```

- To execute dbintegritycheck on Windows against the server instance named myServer, set the reporting level to full, and have the tool write the results to XML files, specify the following command:

  ```
  dbintegritycheck.bat -s myServer -l full -x
  ```

- To execute dbintegritycheck on UNIX against the default server instance and write the output to a directory that you specify, use the following command:

  ```
  ./dbintegritycheck -o /opt/softwareag/MWS/tools/diagnostics/bin/output
  ```

- To execute dbintegritycheck on UNIX against the default server instance, set the reporting level to full, and have the tool write the results to XML files, specify the following command:

  ```
  ./dbintegritycheck -l full -x
  ```