

Chapter 38: Use of Compile Scripts in Table

Files

Compile scripts can be used in table files to preprocess actions, thus speeding up considerably the time it takes users to execute the actions. We describe the use of compile scripts in this chapter.

38.1 Overview

Generally, when a **UPS** command is issued, if **UPS** finds a corresponding action in the product instance's table file, the listed functions get executed. If this function list is lengthy, the command may take a long time to execute. To speed up execution in these cases, **UPS v4** supports the preprocessing of actions in compile scripts. When you preprocess, you run the list of functions once, store the output in a script, and then when the command is later executed, the script is run instead of the functions.

This mechanism can be used for any **UPS** command, but it was developed with the **setup** command in mind. If a **setup** command must setup many, many required products, reading all the files for instance matching can be slow. By use of a compile script, the files can be read once, instead of each time a user runs **setup** on the product.

38.2 Usage Information

The use of compile scripts is most easily explained using an example. The (partial) table file below creates a compile script for the **setup** command when the product instance gets configured. Alternatively, since **ACTION=COMPILE** is defined, you could manually run the command **ups compile** to create the script. The functions listed are described in section 35.3 *Function Descriptions*:

```
ACTION=CONFIGURE
    exeActionRequired("COMPILE")
```

```
ACTION=COMPILE
```

```
writeCompileScript("SETUP", "/my/compile/script")
```

```
ACTION=SETUP
sourceCompileReq("/my/compile/script")
doDefaults()
setupRequired("dog v2_0")
setupRequired("cat v1_1")
...long list...
setupRequired("mouse v3_9")
```

This table file performs the following actions:

- 1) When the product instance is configured (via ACTION=CONFIGURE, which is usually run as part of **ups declare**), the function **exeActionRequired("COMPILE")** runs the functions under ACTION=COMPILE.
- 2) The function **writeCompileScript("SETUP", "/my/compile/script")** under ACTION=COMPILE executes a single operation: run **setup**, and write the output of the setup actions to the script `/my/compile/script`. This **writeCompileScript** function executes all the functions under ACTION=SETUP except the first one (**writeCompileScript** knows to ignore **sourceCompileReq**), and outputs the results to the script `my/compile/script`. For example, for each **setupRequired** line, it completes all the instance matching, and outputs the matched instance to the script.

Later, when **setup** is run by a user, the first function under ACTION=SETUP is executed (**sourceCompileReq ("my/compile/script")**), and the remaining functions are ignored. Therefore, none of the file reads have to occur during normal product setup.



The compiled script will contain hard-coded paths to the instances that were in effect when the script was created. If any product version, root directory or table file changes, the script must be recompiled for **setup** to work properly. Use **ups depend** to determine what the current dependencies are.