

Chapter 30: Chain Files

UPS/UPD supports *chains* to product versions, and chain information is maintained in chain files. In this chapter we describe chain files and how they interact with version files.

30.1 About Chain Files

Chains for a product are maintained in chain files which reside in the product-specific directory under the **UPS** database directory. There is one chain file for each chain name, and it is named according to the chain name, with a suffix of `.chain`, e.g., `current.chain`. A chain file is automatically created by **UPS** the first time an instance of a product is declared with some chain. When any other instances of the same product (regardless of version) get declared with the same chain, one of two things happens:

- a new entry is created in the same chain file, or
- if an entry with the same flavor and qualifiers already exists, the pre-existing entry gets unchained and the new one is chained in its place.

Chain files get created and modified via the **ups declare <chainFlag>** command. A chain file's contents are simply a formatted list of the product instances that were declared with that chain, where each product instance is specified via a set of keywords. When a chain is used in a **UPS/UPD** command, **UPS** looks in the corresponding chain file to match the instance and thus locate the appropriate version file. As discussed in section 27.2 *Instance Matching within Selected Database*, the version file entry locates the product root directory and table file to retrieve the instance.

In **UPS/UPD** commands, the command line option associated with a particular chain can be used in specifying the product instance to match. Using chains is optional, but recommended. Both chained and unchained instances of a product may be declared to **UPS**; the user can still retrieve any instance, chained or not, by specifying its version number.

30.2 Keywords Used in Chain Files

This is a subset of the list given in section 28.4 *List of Supported Keywords*.

Keyword and Default Value (if any)	Description and Notes (if any)
CHAIN	chain name
DECLARED Default: current date and time	the date/time that the instance was declared to UPS or declared with a chain Note: often has multiple values, one for each declaration (e.g., for subsequent chain declarations)
DECLARER Default: current user	userid of user that performed the declaration Note: often has multiple values, one for each declaration (e.g., for subsequent chain declarations)
DESCRIPTION	product description
FILE	type of file (possible values: DBCONFIG, UPDCONFIG, CHAIN, VERSION, TABLE)
FLAVOR	product instance flavor Note: To easily accommodate flavor-neutral setup functions in a table file, FLAVOR can take the value ANY, but <i>only</i> in a table file.
MODIFIED Default: Current date/time	last time the associated instance was changed Note: often has multiple values, one for each declaration/modification (e.g., for subsequent chain declarations)
MODIFIER Default: Current user	userid of user that modified the instance Note: often has multiple values, one for each declaration/modification (e.g., for subsequent chain declarations)
PRODUCT	product name
QUALIFIERS	additional instance specification information often used to indicate compilation options used by developer Notes: appears immediately after a FLAVOR in these files, and is coupled with it to complete the instance identification (see 27.2.3 <i>Qualifiers: Use in Instance Matching</i>)
UPS_DB_VERSION	UPS database version
VERSION	product version

30.3 Chain File Examples

30.3.1 Sample chain file for exmh v1_6_6

This file points to the instance used in the version file of section 29.3.1 *Sample Version File for exmh v1_6_6*. The file `$PRODUCTS/exmh/current.chain` contains the text:

```
FILE = chain
PRODUCT = exmh
CHAIN = current

#*****
#
FLAVOR = SunOS+5
QUALIFIERS = ""
VERSION = v1_6_6
DECLARED = 1998-03-30 21.06.59 GMT
DECLARER = stolz
MODIFIED = 1998-03-30 21.06.59 GMT
MODIFIER = stolz
```

If the given instance hadn't been initially declared as current (as in the command in section 29.3.1), then to create this chain file you would need to declare the instance current, e.g.,:

```
% ups declare -c exmh v1_6_6
```

30.3.2 Sample chain file for foo v2_0

This example illustrates the use of qualifiers. It points to both of the instances in the version file for **foo** in section 29.3.2 *Sample version file for foo v2_0*. That version file will also get modified when these chains are declared. The DECLARER, DECLARED, MODIFIER and MODIFIED fields will include information for the chain declarations.

Making the “current” Chain Declarations

In order for this chain file to have the contents shown below, the following two commands need to be issued:

```
% ups declare -cq superoptimize -f IRIX foo v2_0
% ups declare -cf OSF1 foo v2_0
```

The file `$PRODUCTS/foo/current.chain` contains the text:

```
FILE = CHAIN
```

```

PRODUCT = foo
CHAIN = CURRENT
#
#-----
#
FLAVOR = IRIX
QUALIFIERS = "superoptimize"
    VERSION = v2_0
    DECLARER = aheavey
    DECLARED = 1998-04-15 16.37.58 GMT
    MODIFIED = 1998-05-19 21.06.59 GMT
    MODIFIER = aheavey

FLAVOR = OSF1
QUALIFIERS = ""
    VERSION = v2_0
    DECLARER = aheavey
    DECLARED = 1998-04-15 16.39.58 GMT
    MODIFIED = 1998-05-24 21.06.59 GMT
    MODIFIER = aheavey

```

Sequence of Events at Setup Time

For this example in the IRIX case, the sequence of events upon issuing the command:

```
% setup -q superoptimize foo
```

would be as follows:

- 1) match the FLAVOR (IRIX) and the QUALIFIERS (superoptimize) in this chain file
- 2) find the version (v2_0) and open the corresponding version file (v2_0.version)
- 3) locate the table file (\$FOO_DIR/ups/v2_0.table) and open it
- 4) find the ACTION=SETUP line in the table file and execute the listed functions (if no ACTION=SETUP line is present, **UPS** executes the default **setup** functions)