opentext™

Project and Portfolio Management Center

Software version: All versions

Generating Fiscal Periods

Go to Help Center online

https://admhelp.microfocus.com/ppm/

Document release date: January 2024

Send Us Feedback

Let us know how we can improve your experience with the Generating Fiscal Periods.

Send your email to: admdocteam@opentext.com

Legal Notices

Contents

About this document

This document provides details on how to generate fiscal periods when working with OpenText ™ PPM.

Get started with generating fiscal periods

Introduction to Fiscal Calendars

Different companies use different fiscal years and different fiscal periods within those years. Most fiscal years are based on the familiar Gregorian calendar having twelve months from January to December, with 28 to 31 days per month. However, fiscal years usually begin on a day other than January 1. For example, 's fiscal years run from November 1 to October 31. Moreover, fiscal years can begin on a day other than the first day of one of the months.

Some companies use *standard retail* calendars, which still have twelve periods per year, but usually a total of 52 weeks (364 days) in four 13-week quarters. Other companies use *non-standard retail* calendars, which can have more than or fewer than twelve periods per year, and where the periods, quarters, half-years, and years can all vary in length. For information about standard and non-standard retail calendars, see "Generating Periods for Retail Calendars" on page 26.

Introduction to Configuring and Generating Fiscal Periods

To generate fiscal periods that reflect your company's fiscal calendar and that have the appearance you want, PPM provides configuration files and the `kGenFiscalPeriods.sh` script, allowing you to do the following:

- Shift fiscal years so that they always start on the first day of the month you specify. Quarters are automatically adjusted accordingly.

- Specify the start day of the week, used in Financial Management. The default is Sunday.
- For each supported language, independently specify the following:
  - The formats of periods (years, half-years, quarters, and months) as they appear in financial summaries and elsewhere, and the format of weeks as they appear in the Analyze Cumulative Cost Metrics page in Financial Management
  - The names of months (or similar periods used in retail calendars)
- To reflect retail calendars, specify particular start and end dates of fiscal years, half-years, quarters, and months.

Note: For your convenience, PPM version 9.50 provides a set of fiscal periods in the system default language for the years 1998–2018. Each fiscal year begins on January 1.

Scope of Changes to Fiscal Periods

After you have modified the configuration files to display period names and formats as desired, you run the `kGenFiscalPeriods.sh` script to generate periods for a range of years. The periods appear in or are used for the following:

- Total costs, benefits, and approved budgets for fiscal quarters and fiscal years in financial summaries
- Scenario comparisons in Portfolio Management
- Staffing profiles in Resource Management
- Calendar autocomplete fields, used in creating projects, for example
- Portlets and reports

Note: The periods used in time sheets in Time Management are independent of the configuration of periods described in this document.

Running the Script to Generate Periods

Introduction to the Script and Associated Configuration Files

You run the `kGenFiscalPeriods.sh` script to add sets of periods for a range of years you specify. The periods can cover dates in the past, in the future, or both.

Note: For your convenience, PPM version 9.50 provides a set of fiscal periods in the system default language for the years 1998–2018. Each fiscal year begins on January 1.

Note: The periods used in time sheets in Time Management are independent of the configuration of periods described in this document.

Before running the script, you can optionally modify the following configuration files the script uses as it generates periods:

- `periods.conf` configuration file, which specifies the following for Gregorian calendars:
  - Start month of fiscal years (for all supported languages). For information about changing the start month of fiscal years, see "Shifting the Starting Month of Fiscal Years" on page 9.

  - Start day of each week (for all supported languages), which is used in the Analyze Cumulative Cost Metrics page and portlet in Financial Management. For information about changing the start day of each week, see "Changing the Start Day of Weeks" on page 12.

    Caution: If you need to change the start day of the week from its default (Sunday) for a new installation of PPM, you must change it before generating any periods and you must retain the new setting thereafter. See "Changing the Start Day of Weeks" on page 12.

- For each language installed in PPM, a separate language configuration file named `periods_<Language>.conf`. For example:
  - `periods_en.conf` is the language configuration file for English.
  - `periods_de.conf` is the language configuration file for German.
  - `periods_ko.conf` is the language configuration file for Korean.

  In each language configuration file, you can optionally change the following:
  - Default long and short *names* of the months that appear in that language in PPM. For more information, see "Changing Names of Months" on page 13.
  - Default long and short *formats* of the periods—years, quarters, months, and weeks—that appear in that language in PPM. For more information, see "Changing Formats of Periods" on page 15.
- For *standard or non-standard retail calendars only,* a `.csv` file in which you specify the date ranges for all the periods. For more information, see "Generating Periods for Retail Calendars" on page 26.

In addition, so that users see period names in PPM in the language they select when they log in, you use the script to establish ongoing generation of periods for some or all languages that are installed in PPM (other than the system default language). First, you change the language configuration files for the new languages for which you want to generate periods. Then you run the script with an argument that specifies the new languages, which generates periods for them that span the same time range as the existing periods for other languages. Thereafter, specifying a range of years when you run the script generates new periods for that

range in all the languages that now have periods established. For information about enabling the display of periods in new languages, see "Generating Periods in Additional Languages" on page 20.

Before generating new periods, configure the `periods.conf` and `periods_<Language>.conf` files as needed (see the following sections). If you do not need to modify any of these files, go to "Generating Periods" on page 19.

For more information about using multiple languages in a single installation of PPM, see the *Multilingual User Interface Guide*.

Shifting the Starting Month of Fiscal Years

If your company's fiscal year does not start on January 1, you can run the script to change the starting month for previously generated years and for years to be generated later, so that all years match the fiscal year. Usually, this procedure will need to be performed only once, if at all. For information about the affected functions in PPM, see "Scope of Changes to Fiscal Periods" on page 6.

Note: If any of the fiscal months in your fiscal year do not begin on the first day of the month, use the procedure in "Generating Periods for Retail Calendars" on page 26 to configure the months.

Note: This procedure changes period data in the database. We recommend that you back up the configuration file before modifying it.

To change the starting month for fiscal years:

1. Open the `periods.conf` configuration file in the `<PPM_Home>/conf/fiscal` directory. (The parameters in this file apply to all languages.)
2. Change the `START_MONTH` parameter to a number that represents the month the fiscal year starts, for example `11` for November.
3. Set the `IS_START_MONTH_FOR_NEXT_FISCAL_YEAR` parameter to `true` or `false` according to the relationship between fiscal years and calendar years, as follows:

If the redefined start month starts the fiscal years that are given the same numbers as the *following* calendar years, for example, if November is the redefined start month and November 2009 is the start of fiscal year 2010, set the `IS_START_MONTH_FOR_NEXT_FISCAL_YEAR` parameter to `true`. In this example, if periods for only year 2010 have been generated, the script does the following:

- Shifts fiscal year 2010 to cover November 2009 through October 2010.
- Generates a complete fiscal year 2011 from November 2010 through October 2011, retaining the previously generated months of November and December 2010 (including their monthly data, if any).
- Shifts the quarters of fiscal year 2010 (FY2010), as follows:
  - Q1 of FY2010 becomes November 2009 through January 2010.
  - Q2 of FY2010 becomes February 2010 through April 2010.
  - Q3 of FY2010 becomes May 2010 through July 2010.
  - Q4 of FY2010 becomes August 2010 through October 2010.

If the redefined start month starts the fiscal years that are given the same numbers as the *current* calendar years, for example, if March is the redefined start month and March 2010 is the start of fiscal year 2010, set the `IS_START_MONTH_FOR_NEXT_FISCAL_YEAR` parameter to `false`. In this example, if periods for only year 2010 have been generated, the script does the following:

- Shifts fiscal year 2010 to cover March 2010 through February 2011.
- Generates a complete fiscal year 2009 from March 2009 through February 2010, retaining the previously generated months of January and February 2010 (including their monthly data, if any).
- Shifts the quarters of fiscal year 2010 (FY2010) as follows:
  - Q1 of FY2010 becomes March 2010 through May 2010.
  - Q2 of FY2010 becomes June 2010 through August 2010.
  - Q3 of FY2010 becomes September 2010 through November 2010.
  - Q4 of FY2010 becomes December 2010 through February 2011.

If more than one year has been previously generated, similar changes are made to periods of all the years to ensure that each fiscal year is complete and that all previously generated periods are included in a fiscal year.

4. Save and close the `periods.conf` configuration file.
5. On the PPM Server, navigate to the directory that contains the `kGenFiscalPeriods.sh` script:

   `cd <PPM_Home>/bin`

   where `<PPM_Home>` represents the path where your PPM instance is installed.
6. Run the following script:

   `sh ./kGenFiscalPeriods.sh shift`

   Note: Do not attempt to simultaneously generate additional periods and use the `shift` option in the script.

   The script saves the revised set of periods in a preview file `fiscal_periods_<time stamp>.csv` in the `<PPM_Home>/bin/fiscal/output` directory for you to verify before you commit the periods to the database.
7. Verify that the periods listed in the preview file are appropriate.
8. When you are satisfied with the data in the preview file, run the script again with the `shift` and `commit` options to apply your changes to the database.

   When the script is run to generate periods after you change the `periods.conf` configuration file, the new starting month applies to all previously generated fiscal years and to new fiscal years the script generates. (Do not use the `shift` option in the script when you generate periods.)

   The adjusted fiscal years and quarters apply to displays of financial data in all languages.
9. Return to "Introduction to the Script and Associated Configuration Files " on page 7 to determine whether you need to change configuration files further. If not, go to "Generating Periods" on page 19.

Changing the Start Day of Weeks

In Financial Management, the Analyze Cumulative Cost Metrics page and portlet present projected and actual project costs and other cost metrics cumulatively by week (for more information, see the *Financial Management User Guide*). You can run the script to change which day of the week is used as the start day for the weeks. Usually, this procedure will need to be performed only once, if at all.

Caution: If you need to change the start day of the week from its default (Sunday) for a new installation of PPM, you must change it before generating any periods and you must retain the new setting thereafter.

To change the start day for all of the weeks:

1. Open the `periods.conf` configuration file in the `<PPM_Home>/conf/fiscal` directory. (The parameters in this file apply to all languages.)
2. Change the value of the `START_DAY_OF_WEEK` parameter to one of the following values:
   - `1` to make Sunday the start day for all weeks. This is the default.
   - `2` to make Monday the start day for all weeks.
   - `3` to make Tuesday the start day for all weeks.
   - `4` to make Wednesday the start day for all weeks.
   - `5` to make Thursday the start day for all weeks.
   - `6` to make Friday the start day for all weeks.
   - `7` to make Saturday the start day for all weeks.
3. Save and close the `periods.conf` configuration file.

   You do not need to run the script immediately. The next time the script is run to generate periods, the new starting day of the week will apply to new weeks the script generates.
4. Return to "Introduction to the Script and Associated Configuration Files " on

page 7 to determine whether you need to change configuration files further. If not, go to "Generating Periods" on page 19.

Changing Names of Months

Period names are displayed many places in PPM (see "Scope of Changes to Fiscal Periods" on page 6), as in the table heading rows in the example in " Figure 2-1. Example of period names" below (part of a financial summary), where the names of the quarters are **Q1 2013, Q2 2013,** and **Q3 2013,** and the names of the months are **Jan 13, Feb 13,** and so on.

Figure 2-1. Example of period names

Cost Details(x $1,000)

View: Totals Only | 2 Detail Lines Forecast Only | Forecast and Actuals Months | Quarters | Years | Totals

	Q1 2013						Q2 2013						Q3 2013				
	Jan 13		Feb 13		Mar 13		Apr 13		May 13		Jun 13		Jul 13		Aug 13		
	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Foreca
Capital Total	100.000	86.000	120.000	87.000	110.000	88.000	110.000	90.000	100.000	99.000	100.000	99.000	100.000	97.000	105.000	98.000	100.0
Quarter Total					Forecast: $330.000 Actual: $261.000						Forecast: $310.000 Actual: $288.000						Foreca Actu
	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Forecast	Actual	Foreca
Operating Total	850.000	770.000	890.000	800.000	780.000	770.000	800.000	700.000	800.000	750.000	800.000	750.000	800.000	760.000	880.000	800.000	889.0
Quarter Total					Forecast: $2,520.000 Actual: $2,340.000						Forecast: $2,400.000 Actual: $2,200.000						Forecas Actua
Month Total	950.000	856.000	1,010.000	887.000	890.000	858.000	910.000	790.000	900.000	849.000	900.000	849.000	900.000	857.000	985.000	898.000	989.0
Quarter Total					Forecast: $2,850.000 Actual: $2,601.000						Forecast: $2,710.000 Actual: $2,488.000						Forecas Actua

Edit Costs Add Notes

To change the specific long and short names that are displayed for each of the twelve months of the year, before you run the script to add periods, you modify the `periods_<Language>.conf` language configuration files as desired for each installed language. The script refers to these configuration files as it generates periods.

When the script is run to generate periods after you change any language configuration files, the changes are applied to all previously generated periods and to the new periods the script generates. For information about the affected functions in PPM, see "Scope of Changes to Fiscal Periods" on page 6.

Note: This procedure applies to non-retail calendars, where each fiscal year has twelve Gregorian calendar months and starts on the first day of the same particular month each year.

To support retail calendars, which do not have these limitations, the configuration file for each language includes long and short names for configurable *periods* (for English, `LONG_NAME_PERIOD_1` through `LONG_NAME_PERIOD_12`, and `SHORT_NAME_PERIOD_1` through `SHORT_NAME_PERIOD_12`). These period names are not to be used in this procedure. For more information about retail calendars, see "Generating Periods for Retail Calendars" on page 26.

Note: This procedure changes period data in the database. We recommend that you back up the configuration files before modifying them.

To change the names of the months for a language:

1. Open the `periods_<Language>.conf` configuration file for the language in the `<PPM_Home>/conf/fiscal` directory. See the *System Requirements and Compatibility Matrix* for supported languages and the language code to use in the file name `periods_<Language>.conf`.

   Caution: Do not delete any of the parameters in the `periods_<Language>.conf` file or change their names. Change only their values, as needed.

2. Copy and paste the set of names to be changed, then comment out the original set by typing `#` at the beginning of each line.
3. Change the names of each month as desired.

   For example, using the English configuration file `periods_en.conf`, if you want long names for months to be in all capital letters rather than the default mixed case, and you want short names to have a period at the end, change the copied lines for each month to the following:

```
LONG_NAME_MONTH_1=JANUARY
LONG_NAME_MONTH_2=FEBRUARY
    .
    .
    .
```

```
LONG_NAME_MONTH_12=DECEMBER
SHORT_NAME_MONTH_1=Jan.
SHORT_NAME_MONTH_2=Feb.
   .
   .
   .
SHORT_NAME_MONTH_12=Dec.
```

4. Save and close the `periods_<Language>.conf` configuration file.
5. We recommend that you perform step 1 through step 4 each configured language at this time, changing the month names as desired.

   However, any time after you run the script to add the periods, you can update the language configuration files and run the script with the `-language` argument to change existing month names for the language or languages you specify. For more information, see "Generating Periods in Additional Languages" on page 20.
6. Return to "Introduction to the Script and Associated Configuration Files " on page 7 to determine whether you need to change configuration files further. If not, go to "Generating Periods" on page 19.

Changing Formats of Periods

You can change the short format that will be used to display years and the long and short formats that will be used to display all the other types of periods—quarters, months, and (for Financial Management) weeks. To do so, before you run the script to add periods, you modify the `periods_<Language>.conf` configuration files as desired for each installed language. The script refers to these configuration files as it generates periods.

When the script is run to generate periods after you change any language configuration files, the changes are applied to all previously generated periods and to the new periods the script generates. For information about the affected functions in PPM, see "Scope of Changes to Fiscal Periods" on page 6.

The formats of periods in each `periods_<Language>.conf` file are specified using the following:

- The tokens shown in "Table 2-1. Tokens for formatting periods" below
- Text characters such as the following:
  - Q for quarter
  - The slash (/) to separate month and day

Table 2-1. Tokens for formatting periods

Token	Usage	Example of Resolved Token
{cccc}	Long format for calendar year	2010
{cc}	Short format for calendar year	10
{yyyy}	Long format for fiscal year	2010
{yy}	Short format for fiscal year	10
{seq}	Sequence number	3, for the third in a sequence, as in the third quarter of the year (Q3)
{month}	Long text format for month	September
{mon}	Short text format for month	Sep
{mm}	Numeric format for month	09, for the ninth month of the year
{dd}	Numeric format for day	15, for the 15th day of the month

The default period formats are shown in "Table 2-2. Default period formats" on the next page, and they are the same for all languages in the various `periods_<Language>.conf` files. You can configure the same or different period formats for different languages.

Table 2-2. Default period formats

Period Name	Default Format	Example
FORMAT_YEAR_SHORT_NAME	{yy}	10
FORMAT_HALF_YEAR_LONG_NAME [a]	H{seq} {yyyy}	H2 2010
FORMAT_HALF_YEAR_SHORT_NAME [a]	H{seq} {yy}	H2 10
FORMAT_QUARTER_LONG_NAME	Q{seq} {yyyy}	Q3 2010
FORMAT_QUARTER_SHORT_NAME	Q{seq} {yy}	Q3 10
FORMAT_MONTH_LONG_NAME	{month} {cccc}	September 2010
FORMAT_MONTH_SHORT_NAME	{mon} {cc}	Sep 10
FORMAT_WEEK_LONG_NAME	{mm}/{dd}	09/15
FORMAT_WEEK_SHORT_NAME	{mm}/{dd}	09/15

a. Half-years are not used in PPM at this time.

Note: The long format for years is {yyyy} and it cannot be changed, so it is not available to modify.

The default long format for month names displayed in PPM includes calendar years in the 4-digit format {cccc}. The default short format for month names includes calendar years in the 2-digit format {cc}. You can change these formats to use fiscal years {yyyy} or {yy}.

Note: This procedure changes period data in the database. We recommend that you back up the configuration files before modifying them.

To change the short format for years, or the long or short formats for half-years, quarters, months, or weeks, for a language:

1. Open the `periods_<Language>.conf` configuration file for the language in the `<PPM_Home>/conf/fiscal` directory. See the *System Requirements and Compatibility Matrix* for supported languages and the language code to use in the file name `periods_<Language>.conf`.

> **Caution:** Do not delete any of the parameters in the `periods_<language>.conf` file or change their names. Change only their values, as needed.

2. Copy and paste the formats to be changed (see "Table 2-2. Default period formats" on the previous page), then comment out the original formats by typing `#` at the start of each one to be changed.
3. Use the tokens in "Table 2-1. Tokens for formatting periods" on page 16 and text characters to modify the period formats as desired.

   For example, if you want to display all four digits of the year wherever month and year appear, even when the month uses its short text format, change the copied line:

   ```
   FORMAT_MONTH_SHORT_NAME={mon} {cc}
   ```

   to:

   ```
   FORMAT_MONTH_SHORT_NAME={mon} {cccc}
   ```

   In this way, for example, a display that would have been Sep. 10 by default becomes Sep 2010, which would not be misconstrued as September 10th in any context.

   As another example, if you want to use hyphens instead of slashes in the long format for weeks, replace the “/” text character with the “-” text character to change the copied line:

   ```
   FORMAT_WEEK_LONG_NAME={mm}/{dd}
   ```

   to:

   ```
   FORMAT_WEEK_LONG_NAME={mm}-{dd}
   ```

4. Save and close the `periods_<language>.conf` configuration file.
5. We recommend that you perform step 1 through step 4 for each configured language at this time, changing the period formats as desired.

   However, any time after you run the script to add the periods, you can update the language configuration files and run the script with the `-language` argument to change existing period formats for the language or languages you specify.

For more information, see "Generating Periods in Additional Languages" on the next page.

6. Return to "Introduction to the Script and Associated Configuration Files " on page 7 to determine whether you need to change configuration files further. If not, proceed to "Generating Periods" below.

Generating Periods

To run the script to generate periods:

1. On the PPM Server, navigate to the directory that contains the `kGenFiscalPeriods.sh` script:

   ```
   cd <PPM_Home>/bin
   ```

   where `<PPM_Home>` represents the path where your PPM instance is installed.

2. Run the following script:

   ```
   sh ./kGenFiscalPeriods.sh -startYear <yr1> -endYear <yr2>
   ```

   where `<yr1>` and `<yr2>` are years specified with four digits.

   For example, if you want to add periods from fiscal year 2007 to fiscal year 2025, run the following script:

   ```
   sh ./kGenFiscalPeriods.sh -startYear 2007 -endYear 2025
   ```

   Note: If part or all of the specified range of fiscal years exists, the existing range is not regenerated.

   If generating the specified range of periods would create a time gap between that range and the existing set of periods, the script also generates periods to fill that gap.

   The cumulative set of generated time periods cannot exceed 65 years (up to 15 years in the past and up to 50 years in the future).

   The script saves the generated periods in a preview file `fiscal_periods_<time stamp>.csv` in the `<PPM_Home>/bin/fiscal/output` directory for you to verify before you commit the periods to the database.

3. Verify that the periods listed in the preview file are appropriate.
4. When you are satisfied with the data in the preview file, run the script again for the desired range of years but add the `commit` option to add the periods to the database.

   Periods are added for the system default language and for all the other languages that have been configured using the `kGenFiscalPeriods.sh` script. For information about configuring languages so that the script generates periods in those languages, see "Generating Periods in Additional Languages" below.

Generating Periods in Additional Languages

Note: For your convenience, PPM version 9.50 provides a set of fiscal periods in the system default language for the years 1998–2018. Each fiscal year begins on January 1.

You can use the `kGenFiscalPeriods.sh` script to create periods for additional languages that are installed on PPM. Month names and period formats are as specified in the language configuration files. Each user will view the periods in the session language the user selects upon logging in. The periods the script generates for the additional languages cover the same time span as the periods covered for existing languages.

To add periods in additional installed languages:

1. Specify the desired configurations of month names and period formats for the additional languages in their `periods_<Language>.conf` configuration files. See "Changing Names of Months" on page 13 and "Changing Formats of Periods" on page 15.
2. Run the following script to generate periods for the languages you specify:

   `sh ./kGenFiscalPeriods.sh -language <lang1,lang2,...>`

   where <*lang1,lang2,...*> represents language codes for the set of languages for which you want to generate periods, assuming the language is installed in PPM. See the *System Requirements and Compatibility Matrix* for supported languages

and their language codes. The same language codes are used in the names of the language configuration files the script uses, for example, `de` in `periods_de.conf` for German.

You can specify any combination of languages, separated by commas with no spaces. For example, to add German and Korean, run the following script:

```
sh ./kGenFiscalPeriods.sh -language de,ko
```

Rather than specifying particular languages, you can generate periods for all the supported languages in PPM by running the following script:

```
sh ./kGenFiscalPeriods.sh -language all
```

When you run the script, periods are created in each language you specify, using the corresponding `periods_<language>.conf` configuration file.

The script saves the generated periods in preview files named `fiscal_periods_<language>_<time stamp>.csv` where <*language*> represents the same value as used for the `-language` argument. The preview files, one for each language, are saved in the `<PPM_Home>/bin/fiscal/output` directory for you to verify before you commit the periods to the database.

3. Verify that the periods listed in the preview files are appropriate.
4. When you are satisfied with the data in the preview files, run the script again for the desired languages, but add the `commit` option to add the periods to the database.
5. Restart the PPM Server

Any time after you generate the periods, if you need to change existing month names or period formats for a particular language, you can update its language configuration file and run the script with the `-language` argument and the value for that language.

For more information about using multiple languages in a single installation of PPM, see the *Multilingual User Interface Guide*.

Summary of the kGenFiscalPeriods.sh Script

The following sections summarize the arguments and options for the `kGenFiscalPeriods.sh` script. The script must be run with an argument, or with both the `-startYear` and `-endYear` arguments.

Arguments

"Table 2-3. Arguments for the kGenFiscalPeriods.sh script, continued" on page 24 lists the arguments for the `kGenFiscalPeriods.sh` script.

Table 2-3. Arguments for the kGenFiscalPeriods.sh script

Argument	Description
-endYear <*year*>	Generates additional periods, ending with the fiscal year you specify as a 4-digit year. Used in conjunction with the `-startYear` argument. If part or all of the specified time period exists, it is not regenerated. If generating the specified range of periods would create a time gap between that range and the existing set of periods, the script also generates periods to fill that gap. The cumulative set of generated time periods cannot exceed 65 years (up to 15 years in the past and up to 50 years in the future). Using the commit option with this argument saves new periods to the database.

Table 2-3. Arguments for the kGenFiscalPeriods.sh script, continued

Argument	Description
-export <*file name*>.csv	Exports period definitions (except weeks) from the database to the specified `<file name>.csv` file in the `<PPM_Home>/bin/fiscal/output` directory. After the export, the file contains the following columns of data (for more information, see step 1 in Modify the Period Definitions File): • **Database ID** • **Start Date** • **End Date** • **Period Type** • **Long Name** • **Short Name** Using the commit option with this argument has no effect.
-import <*file name*>.csv	Imports to the database the period definitions from the specified `<file name>.csv` file in the `<PPM_Home>/bin/fiscal/input` directory. You can create a new file to import by copying and modifying an existing exported file. The file to be imported must have the first four columns listed in this table for the export argument. Subsequent columns are ignored. Using the commit option with this argument saves new periods to the database.
-language <*lang1,lang2,...*>	Using the month names and period formats specified in the `period_<Language>.conf` files for the languages specified in the argument, generates the same set of periods for those languages as already exist for other languages. See the *System Requirements and Compatibility Matrix* for the values to specify for the languages for which you need to add periods. Multiple languages must be separated by commas and no spaces. Using the commit option with this argument saves new periods in the new languages to the database.

Table 2-3. Arguments for the kGenFiscalPeriods.sh script, continued

Argument	Description
-startYear <*year*>	Generates additional periods, starting from the fiscal year you specify as a 4-digit year. Used in conjunction with the `-endYear` argument. If part or all of the specified time period exists, it is not regenerated. If generating the specified range of periods would create a time gap between that range and the existing set of periods, the script also generates periods to fill that gap. The cumulative set of generated time periods cannot exceed 65 years (up to 15 years in the past and up to 50 years in the future). Using the commit option with this argument saves new periods to the database.

Options

"Table 2-4. Options for the kGenFiscalPeriods.sh script" below lists the options for the `kGenFiscalPeriods.sh` script.

Table 2-4. Options for the kGenFiscalPeriods.sh script

Option	Description
commit	Commits changes to the database. If this option is not specified, any changes you make are not saved to the database.
help	Displays help for the arguments and options.
shift	Checks the `periods.conf` configuration file to determine whether, to which month, and in which direction to shift the start of each fiscal year, and then performs the shift. Do not attempt to simultaneously generate new periods and use the shift option in the script.
truncate	Deletes all periods currently defined in the database. Use with extreme caution (see the warning below this table). To take effect, this option must be used with the commit option.

Caution: Using the truncate option deletes all periods in the system. We recommend using this option only on new installations when you intend to delete the standard fiscal calendars provided by default and then create and import a new set of retail calendar periods, possibly based on an external file.

Generating Periods for Retail Calendars

Introduction to Retail Calendars

Some companies use *standard retail* calendars, which have twelve periods per year, but in most years each quarter has thirteen weeks—most commonly a 4-week period, then a 5-week period, then another 4-week period. Every week starts on a Sunday. Standard retail calendars have a total of exactly 52 weeks (364 days), except that once every five or six years, they have 53 weeks (371 days) to compensate for the actual number of days in a Gregorian calendar year—365 or, for leap years, 366.

Other companies use *non-standard retail* calendars, which can have more than or fewer than twelve periods per year, and where each individual period, quarter, half-year, and year can vary in length.

You can change the periods in fiscal years in PPM so that they reflect standard or non-standard retail calendars instead of the Gregorian calendar.

Requirements for the Period Definitions File

To support retail calendars, PPM uses a period definitions file that provides significant flexibility in the specification of the date ranges for fiscal years, half-years, quarters, and periods. For example, a particular fiscal year can have 11 or 13 periods of various lengths, each typically about one month long. For 12-period retail calendars, in each language's configuration file you specify the following periods:

- `LONG_NAME_PERIOD_1` through `LONG_NAME_PERIOD_12`
- `SHORT_NAME_PERIOD_1` through `SHORT_NAME_PERIOD_12`

The set of dates you specify for each type of period in the fiscal year must meet the following requirements:

- Complete fiscal years must be specified. The 4-digit year number must be the first row of data for each fiscal year defined below it.
- There can never be any gaps or overlaps among the dates that are covered by a series of fiscal years.
- There can never be any gaps or overlaps among the dates that are covered by any type of period (year, half-year, quarter, or month) within any fiscal year.
- All the types of periods in a fiscal year must be fully specified.
- The total span of every pair of adjacent half-years must match the span of the year in which they fall. Similarly, the total span of every pair of adjacent quarters must match the span of the half-years in which they fall. Finally, the total span of a set of adjacent months or similar periods must match the span of the quarters in which they fall.

  For example, following are some of the constraints on the sets of half-years, quarters, and months in a fiscal year that uses a retail calendar and runs from November 1, 2009 to October 30, 2010 (364 days):

  - The half-years could be November 1, 2009–May 1, 2010 and May 2, 2010–October 30, 2010.
  - The half-years could *not* be November 1, 2009–May 1, 2010 and May 1, 2010–October 30, 2010, because of the overlap on May 1.
  - The half-years could *not* be November 1, 2009–April 30, 2010 and May 2, 2010–October 30, 2010, because of the gap on May 1.
  - The first day of the first quarter and the first day of the first month (or similar period) must be November 1, 2009.
  - The last day of the last quarter and the last day of the last month (or similar period) must be October 30, 2010.

Caution: We strongly recommend that you establish periods for retail calendars only on new installations of PPM, that is, installations that have not been upgraded from previous versions or used for production. If you change the start and end dates of months for which existing financial data used non-retail calendars, the data becomes inaccurate and meaningless, because it has been tracked using Gregorian calendar months.

Generating Periods for a Retail Calendar

As detailed in the following sections, you establish periods for a retail calendar for a new installation as follows:

- Delete any existing period definitions in PPM
- Create a period definitions file from scratch or by exporting the period definitions, if any, in the database
- Modify the period definitions file to cover a set of fiscal years with the required sets of periods (years, half-years, quarters, and months), each specified to span particular start and end days
- Modify the language configuration files (`periods_<Language>.conf`) to use the desired period names and formats
- Run the `kGenFiscalPeriods.sh` script to import the period definitions file and generate the set of periods it specifies

Caution: Weeks are generated among the periods even though you *must not* specify them in the period definitions file. If you need to change the start day of each week, you must do so before generating any periods. See "Changing the Start Day of Weeks" on page 12.

Delete Existing Period Definitions

Note: Performing this procedure deletes all periods in the system. We recommend using this option only on new installations when you intend to delete the standard fiscal calendar periods provided by default and then create

and import a new set of retail calendar periods based on an external file.

To delete all existing period definitions from PPM:

1. On the PPM Server, navigate to the directory that contains the `kGenFiscalPeriods.sh` script:

   `cd <PPM_Home>/bin`

   where `<PPM_Home>` represents the path where your PPM instance is installed.
2. Run the following script:

   `sh ./kGenFiscalPeriods.sh truncate commit`

Create the Period Definitions File

To export the current period definitions from PPM into a file you modify, run the following script:

`sh ./kGenFiscalPeriods.sh -export <file name>.csv`

where `<file name>` is the name of the file to which you want to copy the period definitions.

For example, if you want to export the data to a file named `exported_periods.csv`, run the following script:

`sh ./kGenFiscalPeriods.sh -export exported_periods.csv`

The file is saved in the directory `<PPM_Home>/bin/fiscal/output.`

Alternatively, create a period definitions file with a `.csv` extension from scratch. For information about the required columns and data, see "Modify the Period Definitions File" below.

Modify the Period Definitions File

To modify the period definitions file to cover a set of fiscal years with the required sets of periods:

1. Open the file in a text editor such as Notepad. Opening the file in Excel can change the format of dates, making the file invalid for later importing into PPM.

   An exported file contains, or the file you create needs to contain, the following columns of data:

   - **Database ID.** A unique identifier from the database for the period. This column is empty if no periods have been generated, but it must exist when you import the file later. Do not change any entries in this column.
   - **Start Date.** The specific start date of the period in YYYY-MM-DD format.
   - **End Date.** The specific end date of the period in YYYY-MM-DD format.
   - **Period Type.** A string identifying the type of period:
     - A specific 4-digit year such as **2009**
     - **HY** for half-year
     - **Q** for quarter
     - **M** for month

   If period definitions were exported to a `.csv` file, that file also contains the following columns, which are provided for information only and are not needed when you import the file later:

   - **Long Name.** The long name of the period.
   - **Short Name.** The short name of the period.
2. Modify the dates and period types as needed to reflect the retail calendar your company uses. Make sure the constraints described in "Requirements for the Period Definitions File" on page 26 are met. For usability, we recommend that you not move any rows to other positions in the table.

Modify the Language Configuration Files

You configure the period names and formats as desired in each language configuration file (`periods_<Language>.conf`).

You must use the long and short *period* names provided to support retail calendars (listed under the comments indicating that they are imported). For English, these period names are as follows:

- `LONG_NAME_PERIOD_1` through `LONG_NAME_PERIOD_12`
- `SHORT_NAME_PERIOD_1` through `SHORT_NAME_PERIOD_12`

Do *not* use long and short month names LONG_NAME_MONTH_1 through LONG_NAME_MONTH_12 or SHORT_NAME_MONTH_1 through SHORT_NAME_MONTH_12. Keep this requirement in mind as you revise each language configuration file.

If, for example, the fiscal years you want to generate for your retail calendar have up to thirteen periods, you must add and specify `LONG_NAME_PERIOD_13` and `SHORT_NAME_PERIOD_13`. If, on the other hand, the fiscal years you want to generate for your retail calendar have no more than eleven periods, delete `LONG_NAME_PERIOD_12` and `SHORT_NAME_PERIOD_12`.

When you use the default calendar year format {cccc} for period names, the year displayed on the right of the period depends on the calendar year in which the eighth day of the period falls. For example, consider a period from December 26, 2008 to January 22, 2009. Because January is the month in which this period's eighth day falls and January is in calendar year 2009, the period name in this example is displayed as January 2009.

If you rename the periods such that they are not named for months, for example if you specify the short names for periods as Per 1: through Per 12:, you might want the period formats to include fiscal years—format {yyyy}—instead of calendar years. For example, if you specify the period format to include fiscal years and if fiscal year 2009 starts on November 3, 2008, then the period that starts on November 3, 2008 would be shown as Per 1: 2009, reflecting the fiscal year, rather than Per 1: 2008.

To configure period names and formats, see "Changing Names of Months" on page 13 and "Changing Formats of Periods" on page 15.

Import the Period Definitions File

To import the modified period definitions into PPM:

1. Save the modified period definitions file with a `.csv` extension in the directory `<PPM_Home>/bin/fiscal/input.`
2. Run the following script:

   `sh ./kGenFiscalPeriods.sh -import <file name>.csv`

   where <*file name*> is the name of the modified period definitions file in the folder specified in step 1.

   For example, if you want to import the data from the file named `period_definitions.csv`, run the following script:

   `sh ./kGenFiscalPeriods.sh -import period_definitions.csv`

   The script saves the generated periods in a preview file `fiscal_periods_<time stamp>.csv` in the `<PPM_Home>/bin/fiscal/output` directory for you to verify before you commit the periods to the database.
3. Verify that the periods listed in the preview file are appropriate.
4. When you are satisfied with the data in the preview file, run the script with the `-import` argument again but add the `commit` option to add the periods to the database.