Updating Go Installations  
with the  
Latest IANA Time Zone Data

## Table of Contents

[Introduction 1](#_Toc21483512)

[IANA Time Zone Information 1](#_Toc21483513)

[Go Relies on IANA Time Zone Database 1](#_Toc21483514)

[Go Provides for IANA Database Updates 2](#_Toc21483515)

[Existing Go Time Zone Configuration 3](#_Toc21483516)

[Configuration Overview 3](#_Toc21483517)

[Directory/Folder Structure 3](#_Toc21483518)

[The Go Time Directory 3](#_Toc21483519)

[zoneinfo.zip 3](#_Toc21483520)

[update.bash 4](#_Toc21483521)

[Upgrading Go Time Zone Configuration 4](#_Toc21483522)

[Objective - Create New File: *zoineinfo.zip* 5](#_Toc21483523)

[Two Methods for Updating Go Time Zone Data 6](#_Toc21483524)

[Method-1 The Easy Way: ***update.bash*** 6](#_Toc21483525)

[Step-1 “Safety First” 6](#_Toc21483526)

[Step-2 Copy *update.bash* to Scratch Directory *D:\tz* 6](#_Toc21483527)

[Step-3 Modify *update.bash* 7](#_Toc21483528)

[Step-4 Execute *update.bash* Script 8](#_Toc21483529)

[Step-5 Copy zoneinfo.zip to the Go Time Directory 9](#_Toc21483530)

[Method-1: Pro’s and Con’s 10](#_Toc21483531)

[Advantages 10](#_Toc21483532)

[Drawbacks 10](#_Toc21483533)

[Method-2 The Not So Easy Way 11](#_Toc21483534)

[Step-1 “Safety First” 11](#_Toc21483535)

[Step-2 Identify the Latest IANA Time Zone Data Version 11](#_Toc21483536)

[Access the IANA Time Zone Website 11](#_Toc21483537)

[Capture the Latest Time Zone Version Number 11](#_Toc21483538)

[Step-3 Navigate to Scratch Directory 12](#_Toc21483539)

[Step-4 Execute Bash Commands 13](#_Toc21483540)

[Execute Bash Command #1 - Set Environment to Exit on Error 13](#_Toc21483541)

[Execute Bash Command #2 - Delete Old ***work*** directory tree 13](#_Toc21483542)

[Execute Bash Command #3 - Create ***Work*** Directory 13](#_Toc21483543)

[Execute Bash Command #4 - Change to ***Work*** Directory 13](#_Toc21483544)

[Execute Bash Command #5 - Create zoneinfo Directory 14](#_Toc21483545)

[Execute Bash Command #6 - Download Latest Code 14](#_Toc21483546)

[Execute Bash Command #7 - Download Latest Data 14](#_Toc21483547)

[Execute Bash Command #8 - Unzip Code tar 15](#_Toc21483548)

[Execute Bash Command #9 - Unzip Data tar 15](#_Toc21483549)

[Execute Bash Command #10 - Build Time Zone Files 15](#_Toc21483550)

[Execute Bash Command #11 - Change Directory to zoneinfo 16](#_Toc21483551)

[Execute Bash Command #12 - Delete Old **zoneinfo.zip files** 16](#_Toc21483552)

[Execute Bash Command #13 - Zip the **zipinfo** Directory 16](#_Toc21483553)

[Execute Bash Command #14 - Change Directory to **D:\tz** 17](#_Toc21483554)

[Bash Command Wrap-Up 17](#_Toc21483555)

[Step-5 Copy zoneinfo.zip to the Go Time Directory 18](#_Toc21483556)

[Method-2: Pro’s & Con’s 18](#_Toc21483557)

[Advantages 18](#_Toc21483558)

[Drawbacks 18](#_Toc21483559)

[Clean-Up 19](#_Toc21483560)

[Function LoadLocation 20](#_Toc21483561)

## Introduction

### IANA Time Zone Information

The *Go* programming language relies on time zone information provided by the IANA Time Zone database. This database is widely recognized as the leading authority on world time zones.

Reference:

<https://en.wikipedia.org/wiki/List_of_tz_database_time_zones>

<https://en.wikipedia.org/wiki/Tz_database>

The IANA Time Zone data base and reference information is located at:

<https://www.iana.org/time-zones>

<https://data.iana.org/time-zones/releases/>

### Go Relies on IANA Time Zone Database

The *Go* Programming Language uses IANA Time Zones in date-time calculations.

Reference:

<https://golang.org/pkg/time/#LoadLocation>

<https://golang.org/pkg/time/>

### Go Provides for IANA Database Updates

The IANA Time Zone database is updated throughout the year as warranted by time zone modifications initiated by nation states. In order to stay current with time zone modifications, the end-user has the option of updating the local *Go* installation with the latest version of the IANA Database. *Go* does not automatically update the time zone data base. This task is therefore left as manual operation to be performed by the end-user.

## Existing Go Time Zone Configuration

### Configuration Overview

#### Directory/Folder Structure

Go is typically installed on a drive in the ***Go*** directory or folder. A common Windows installation would install ***Go*** on C-Drive as follows:

C:\Go

#### The Go Time Directory

Time zone information is stored in this directory structure under the ***Go\lib\time*** directory. Following the example above, on Windows, this would be styled as follows:

**C:\Go\lib\time**

This ***Go*** directory is also defined by the environment variable $GOROOT. Therefore, the Go time directory could also be defined as:

**$GOROOT/lib/time**

#### zoneinfo.zip

All time zone information is contained within a file residing in the ***Go\Lib\time*** directory named, ***zoneinfo.zip***. Using the example above this would equate to a path, file name, file extension of ***C:\Go\lib\time\zoneifno.zip***.

This path and file name could also be referred to as:

$GOROOT/lib/time/zoneinfo.zip

#### update.bash

This file also resides in the **Go\lib\time** directory (a.k.a. $GOROOT/lib/time). This bash file is used to generate a new ***zoneinfo.zip*** file. Following the example cited above the path, file name and file extension for this file would be styled as follows:

***$GOROOT/lib/time/update.bash***

### Upgrading Go Time Zone Configuration

To upgrade the time zone data for your Go installation, you will need to create a new version of ***zoneinfo.zip*** containing the latest IANA Time Zone information.

## Objective - Create New File: *zoineinfo.zip*

To update time zone information in ***Go*** it is necessary to update the ***zoneinfo.zip*** file described in the previous section.

Therefore, the objective is to create a file named *zoneinfo.zip* and copy it to the appropriate directory thereby replacing the old version of ***zoneinfo.zip***. Once this operation is completed, the new version of ***zoneinfo.zip*** will be used by ***Go*** to correctly process time zones anywhere in the world using the latest IANA time zone data.

When you finish building the latest version of the *zoneinfo.zip* file, make sure it is copied to the ***Go*** time directory:

***$GOROOT/lib/time/zoneinfo.zip***

## Two Methods for Updating Go Time Zone Data

### Method-1 The Easy Way: ***update.bash***

#### Step-1 “Safety First”

1. Make a backup copy of the time zone information file: ***$GOROOT/lib/time/zoneinfo.zip***.
2. Create a new, empty scratch directory which can be used for time zone update operations. Example: ***D:\tz***.

#### Step-2 Copy *update.bash* to Scratch Directory *D:\tz*

1. The File ***update.bash*** resides in the Go time directory: ***$GOROOT/lib/time/update.bash***
2. Copy ***update.bash*** to the new empty, scratch directory created in Step-1, above.

**Copy ->**

From: ***$GOROOT/lib/time/update.bash***

To: ***D:\tz\update.bash***

#### Step-3 Modify *update.bash*

After copying update.bash to the scratch directory, it will be necessary to modify this bash script using a text editor.

1. Point your browser at <https://www.iana.org/time-zones> .
2. Make a note of the latest time zone version. Example: ***2019c***
3. Open the scratch directory version of ***update.bash*** in a text editor (see Step-2 above: ***D:\tz\update.bash***).
4. Locate the bash script section labeled **“# Versions to use.”** in your text editor.
5. Modify the “CODE” and “DATA” variables to reflect the latest time zone version captured in Step-3 a, above.

**From:**

**CODE=2019b**

**DATA=2019b**

**To:**

**CODE=2019c**

**DATA=2019c**

Instead of **“2019c”** you will of course use the latest IANA time zone version.

1. Save the modified ***update.bash*** file and exit your text editor.

#### Step-4 Execute *update.bash* Script

1. Open a terminal window and change directories to the scratch directory (i.e. make the scratch directory the current window.

On windows this means using ***Cygwin***. Change Directory to scratch directory, ***D:\tz***.

**cd /cygdrive/d**

**cd ./tz**

1. Now, execute the bash script.

**./update.bash**

1. If all went well and no errors were emitted, the following final message should be displayed in the terminal window:

**“New time zone files in zoneinfo.zip.”**

1. Close the Terminal Window
2. Assuming a successful bash script execution, only two files should remain in the scratch directory:
3. **update.bash**
4. **zoneinfo.zip**

This ***zoneinfo.zip*** file is the new replacement file configured with the latest IANA Time Zone information.

#### Step-5 Copy zoneinfo.zip to the Go Time Directory

Copy the newly created zoneinfo.zip file from the scratch directory to the Go time directory.

1. After ensuring you have backup copy, delete the old zoneinfo.zip file from the Go Time Directory.

Delete File: ***$GOROOT/lib/time/zoneinfo.zip***

1. Copy the new zoneinfo.zip file from the scratch directory to the Go Time directory.

Copy:

From: ***D:\tz\zoneinfo.zip***

To: **$GOROOT/lib/time/zoneinfo.zip**

1. Done - Mission Accomplished

That’s it. The objective has been achieved. By copying the file **zoneifno.zip** to the Go Time directory **($GOROOT/lib/time)**, updated time zone information is now configured. The Go compiler will now incorporate this updated time zone information into all date/time calculations executed in code.

#### Method-1: Pro’s and Con’s

##### Advantages

It’s easy. Compared to Method-2, creation of the file zoneinfo.zip file requires far fewer manual operations.

##### Drawbacks

Unfortunately, there is no version information contained within ***zoneinfo.zip***. Therefore, you will not be able to confirm the IANA version number from an examination of ***zoneinfo.zip***. However, if the ***CODE*** and ***DATA*** variables were updated correctly as outlined in **Step-3** above, there is a high degree of confidence that the **update.bash** script will execute correctly and the new ***zoneinfo.zip*** file contains the latest time zone data.

### Method-2 The Not So Easy Way

This method involves downloading the IANA Time Zone files and manually building the **zoneinfo.zip** file.

#### Step-1 “Safety First”

1. Make a backup copy of the time zone information file:

***$GOROOT/lib/time/zoneinfo.zip.***

1. Create a new, empty scratch directory which can be used for time zone update operations. Example:

***D:\tz.***

#### Step-2 Identify the Latest IANA Time Zone Data Version

##### Access the IANA Time Zone Website

Point your browser at <https://www.iana.org/time-zones>

##### Capture the Latest Time Zone Version Number

Make a note of the latest time zone version. Example: **2019c**

#### Step-3 Navigate to Scratch Directory

1. Open a Terminal Window.
2. Make the scratch directory the current directory. On windows this means using Cygwin. Change Directory to scratch directory, ***D:\tz***.

**cd /cygdrive/d**

**cd ./tz**

#### Step-4 Execute Bash Commands

After making the scratch directory (***D:\tz***) the current directory, set the environment and create the sub-directories by issuing the following commands. Again, you must be in the scratch directory (***D:\tz***) when issuing these bash commands.

##### Execute Bash Command #1 - Set Environment to Exit on Error

**set -e**

This sets the environment such that scripts will exit on error.

##### Execute Bash Command #2 - Delete Old ***work*** directory tree

**rm -rf work**

This deletes the sub-directory ***work*** and child directories in the tree.

##### Execute Bash Command #3 - Create ***Work*** Directory

**mkdir ./work**

This creates the sub-directory ***work*** (***D:/tz/work***).

##### Execute Bash Command #4 - Change to ***Work*** Directory

**cd ./work**

Change directory to sub-directory **work (D:/tz/work)**.

##### Execute Bash Command #5 - Create zoneinfo Directory

**mkdir ./zoneinfo**

Creates new sub-directory, ***zoneinfo (D:\tz\work\zoneinfo)***

##### Execute Bash Command #6 - Download Latest Code

Replace the text phrase 2019c in the command syntax below with the latest time zone version acquired in Step-2 above.

**curl -L -O https://www.iana.org/time-zones/repository/releases/tzcode2019c.tar.gz**

This **curl** command must be executed on one line from the **work** directory ***(D:\tz\work)***.

##### Execute Bash Command #7 - Download Latest Data

Replace the text phrase 2019c in the command syntax below with the latest time zone version acquired in Step-2 above.

**curl -L -O https://www.iana.org/time-zones/repository/releases/tzdata2019c.tar.gz**

This **curl** command must be executed on one line from the **work** directory (***D:\tz\work***).

##### Execute Bash Command #8 - Unzip Code tar

Replace the text phrase **2019c** in the command syntax below with the latest time zone version acquired in Step-2 above. This command will unzip the file downloaded with Command # 6, above.

**tar xzf tzcode2019c.tar.gz**

This command must be executed on one line from the **work** directory **(D:\tz\work)**.

##### Execute Bash Command #9 - Unzip Data tar

Replace the text phrase 2019c in the command syntax below with the latest time zone version acquired in Step-2 above. This command will unzip the file downloaded with Command # 7, above.

**tar xzf tzdata2019c.tar.gz**

This command must be executed on one line from the **work** directory **(D:\tz\work)**.

##### Execute Bash Command #10 - Build Time Zone Files

This will execute **make** command and create the actual time zone data files.

**make CFLAGS=-DSTD\_INSPIRED AWK=awk TZDIR=zoneinfo posix\_only**

This command must be executed on one line from the **work** directory **(D:\tz\work)**.

##### Execute Bash Command #11 - Change Directory to zoneinfo

Change directory to ***D:\tz\work\zoneinfo***

**cd ./zoneinfo**

If uncertain about your current directory, run bash command, **pwd**. This will display the current directory.

It is important to be in the correct directory.

##### Execute Bash Command #12 - Delete Old **zoneinfo.zip files**

As a precaution, this command is run from ***D:\tz\work\zoneinfo***. It will effectively delete old ***zoneinfo.zip*** files which reside in directory ***D:\tz***, two directories above the current directory. The new ***zipinfo.zip*** file will be created in ***D:\tz***.

**rm -f ../../zoneinfo.zip**

##### Execute Bash Command #13 - Zip the **zipinfo** Directory

**zip -0 -r ../../zoneinfo.zip \***

**“-0”** (Zero) option = “store files with no compression”

This zip command will create the ***zoneinfo.zip*** file containing the specified time zone information downloaded in Commands # 6 and 7, above.

##### Execute Bash Command #14 - Change Directory to **D:\tz**

**cd ../..**

This assumes that the change directory command is being executed from directory, **D:\tz\work\zoneinfo**. After executing this “change directory” command, the current working directory should be ***D:\tz***. If in doubt about the current directory, issue the ***pwd*** command to list the current working directory.

##### Bash Command Wrap-Up

Close the terminal window.

This concludes the bash commands required to create the new ***zoneinfo.zip*** file containing the time zone information downloaded in **Commands # 6 and 7**. If all steps were completed successfully and no errors were encountered, the new ***zipinfo.zip*** file resides in the base scratch directory, ***D:\tz***.

The following steps involve copying this ***zipinfo.zip*** file to the Go time directory **($GOROOT/lib/time)**. After completing this operation, consider taking action to clean up scratch directory ***D:\tz***.

#### Step-5 Copy zoneinfo.zip to the Go Time Directory

Copy the newly created ***zoneinfo.zip*** file from the scratch directory to the Go time directory.

1. After ensuring you have backup copy, delete the old zoneinfo.zip file from the Go Time Directory.

Delete File: **$GOROOT/lib/time/zoneinfo.zip**

1. Copy the new zoneinfo.zip file from the scratch directory to the Go Time directory.

**Copy:**

From: **D:\tz\zoneinfo.zip**

To**: $GOROOT/lib/time/zoneinfo.zip**

#### Method-2: Pro’s & Con’s

##### Advantages

Method-2 provides more granular control of the upgrade process. This increased flexibility allows the end user to specify any version of the time zone database in the release history. In addition, this method allows absolute verification of the version downloaded and processed into the final ***zoneinfo.zip*** file.

##### Drawbacks

Compared to Method-1, Method-2 has many more moving parts, increases the chance of error and is considerably more labor intensive.

## Clean-Up

After completing the procedure, give some thought to documenting the version of time zone data now residing in the Go Time directory ***($GOROOT/lib/time)***.

Each time the $GOROOT directory is updated with a new version of Go downloaded from golang.org, two files in the Go time directory are overwritten: ***$GOROOT/lib/time/zoneinfo.zip*** and ***$GOROOT/lib/time/update.bash***.

In this case, the only source of time zone version numbers is found through an examination of the **CODE** and **DATA** variables contained in ***update.bash***.

Therefore, if you choose to manually update time zone data using one of the two methods outlined above, you may want to include a text file in the Go time directory ***($GOROOT/lib/time)*** documenting if version of time zone data contained in ***zoneinfo.zip***.

## Function LoadLocation

The Go Time Package contains a function named [LoadLocation](https://golang.org/pkg/time/#LoadLocation). This function is used to load time zones for date/time calculations.

func LoadLocation(name string) (\*Location, error)

***LoadLocation*** returns the Location with the given name.

If the name is "" or "UTC", LoadLocation returns UTC. If the name is "Local", LoadLocation returns Local.

Otherwise, the name is taken to be a location name corresponding to a file in the [*IANA Time Zone database*](https://www.iana.org/time-zones), such as "America/New\_York".

The time zone database needed by LoadLocation may not be present on all systems, especially non-Unix systems. LoadLocation looks in the directory or uncompressed zip file named by the ZONEINFO environment variable, if any, then looks in known installation locations on Unix systems, and finally looks in $GOROOT/lib/time/zoneinfo.zip.