IBC USER GUIDE

IMPORTANT

> NOTES REGARDING AUTO-RESTART IN TWS/GATEWAY 1018 and later versions.
>
> Starting with version 3.15.0, IBC now has the ability to allow TWS/Gateway to use the autorestart mechanism originally introduced in version 974/975.
>
> This means that you can now set TWS/Gateway to run all week with a single login at the start of the week, under the control of IBC.
>
> To configure this behaviour use the `AutoRestart` setting in the `Lock and Exit` section of the TWS/Gateway configuration dialog. Alternatively use the `AutoRestartTime` setting in `config.ini`.
>
> This support for auto-restart can lead to confusion if you sometimes want to start TWS/Gateway without using IBC. For advice on this, see the section entitled **How to run TWS/Gateway without IBC when IBC is installed** towards the end of this document.

IMPORTANT

> Make sure you read the information in the **Scope of this User Guide** section.
>
> Note that in the remainder of this document, 'Unix' is used to refer to all Unix-derived operating systems, including Linux and macOS.

Introduction

Overview of IBC

IBC enables Interactive Brokers' Trader Workstation (TWS) and Gateway to be run in 'hands-free' mode, so that a user need not be present. This makes possible the deployment of unattended automated trading systems.

IBC loads TWS or the Gateway and then 'listens' for various events (such as the display of dialogs) that would normally require user intervention. It can then automatically take appropriate action on the user's behalf. For example, it automates the TWS and Gateway login by filling the login dialog with your credentials and 'clicking' the login button.

Here are some of the things IBC does for you:

- starts TWS or the Gateway
- logs you into TWS or Gateway
- clicks the YES button if the "Accept incoming connection?" dialog is displayed
- allows TWS or Gateway to automatically restart each day without need for repeated authentication: authentication is only required the first time during the week that TWS or Gateway run after 01:00 ET on Sunday
- allows two-factor authentication using the IBKR Mobile app, including repeated alerts until the user acknowledges

IBC also responds to certain commands sent to it by another program, for example to tell TWS/Gateway to shut itself down cleanly.

Scope of this User Guide

This User Guide is intended to help you get started with IBC. It does not cover every feature in depth.

Note that the configuration file `config.ini` that governs IBC's behaviour contains extensive notes that provide more information on the various settings.

Acknowledgement

This User Guide has been produced using the Pandoc document conversion system to produce the PDF from the markdown source.

Getting Started

Checklist

Here is a summary of the steps you need to perform to get IBC up and running properly.

1. Install the offline version of Interactive Brokers Trader Workstation (see *Interactive Brokers Trader Workstation* in the *Prerequisites* section), and make sure that it uses the English language setting.

   Please note that you MUST download the OFFLINE version of TWS, not the self-updating version: IBC DOES NOT WORK with the self-updating version of TWS.

2. On Linux, you'll need to install xterm if it isn't already installed (to check, enter the command 'xterm' in a terminal session). Check your Linux documentation for how to install xterm. (Note that xterm is not used on macOS.)

3. Download the appropriate IBC distribution ZIP file for your operating system (see the *Where to get IBC* section).

4. Install IBC (see the *Installing IBC* section). Please note that if you already have an existing IBC installation, it's wise to rename its folder before installing the new version in case you need to revert to it later.

5. Create an encrypted folder called `ibc` in your personal filestore (see *Protecting the Password* in the *Password Security* section).

6. Copy the configuration file (called `config.ini`) from the IBC installation folder to the encrypted folder created in step 4.

7. Edit the `config.ini` file,using a text editor such as Notepad, to set your username and password in the `IbLoginId` and `IbPassword` settings. It's advisable to use your paper-trading credentials at first to check things out, and for this you'll also need to set the `TradingMode` setting.

8. Check that the correct major version number for TWS is set in the shell script files in the IBC installation folder: these files are `StartTWS.bat` and `StartGateway.bat` on Windows, `twsstart.sh` and `gatewaystart.sh` on Unix, `twsstartmacos.sh` and `gatewaystartmacos.sh` on macOS.

   To find the TWS major version number, first run TWS or the Gateway manually using the IBKR-provided icon, then click `Help > About Trader Workstation` or `Help > About IB Gateway`. In the displayed information you'll see a line similar to this:

   ```
   Build 10.19.1f, Oct 28, 2022 3:03:08 PM
   ```

For Windows and Linux, the major version number is 1019 (ie ignore the period after the first part of the version number).

For macOS, the major version number is 10.19. (Note that this is different from the equivalent Windows and Linux settings because the macOS installer includes the period in the install folder name).

Now open the script files with a text editor and ensure that the TWS_MAJOR_VRSN variable is set correctly.

9. At this stage, everything is set up to run IBC with its default settings, which will start TWS and attempt to log it into your paper-trading user. It is worthwhile doing this to check that everything works before further customising it to suit your needs. To do this, run the relevant shell script (`StartTWS.bat` on Windows, `twsstart.sh` on Unix, `twsstartmacos.sh` on macOS) from the IBC installation folder. If everything is satisfactory, shut down IBC by closing TWS in the usual way.

   Note that when you start IBC, information about the startup process is logged to a file to aid diagnosing any faults that may occur. You will be notified of the log file name during the startup sequence. Please include this file when reporting problems with IBC.

10. Now you can edit the configuration file `config.ini` to make any further customisations you need. See *Configuring IBC* for further information.

11. If you did not install TWS and IBC in their default locations, and store the configuration file in the recommended location, you will have to edit the shell scripts in the IBC installation folder accordingly. They contain comments that will help you do this correctly.

12. If you intend to run API programs to connect with TWS, you will need to manually edit the API settings in TWS's Global Configuration Dialog.

13. If you want TWS to automatically restart every day during the week without you having to re-authenticate, you'll need to ensure the AutoRestart time is set appropriately in the Lock and Exit section of the Global Configuration dialog. Note that the only alternative to auto-restart is auto-logoff: this shuts down TWS completely at the specified time, and it's then up to you to restart it and re-authenticate.

Prerequisites

This section details the other software that is needed to run IBC.

Java Runtime

Both IBC and TWS/Gateway are Java programs, and therefore the Java Runtime needs to be accessible, but you don't have to do anything to ensure this.

The TWS and Gateway installers include a hidden version of Java that Interactive Brokers have used for developing and testing TWS. This version also runs IBC perfectly, and the IBC scripts ensure that it is used.

This means that it is not necessary to ensure that Java is installed on your computer. It doesn't matter if it is already installed, but the IBC scripts won't use it. However the scripts do make provision for declaring specifically which Java installation is to be used in exceptional situations where necessary.

If you had previously installed Java for use with old versions of TWS, but do not need it for any other programs, then you might want to consider uninstalling it once you have finished setting up IBC.

Interactive Brokers Trader Workstation

Before running IBC, you will need to download and install the **offline** version of Trader Workstation from the Interactive Brokers website.

The location of the TWS dowloads page on IBKR's website varies from time to time, and from country to country. At the time of writing, on IBKR's US website (linked above) you need to click the `Trading` menu near the top of the page, then select `Platforms`, and then click `Download Software` under the Trader Workstation panel. Currently a valid direct link is TWS Software.

IBKR provides two modes of operation for TWS:

- an online, or self-updating TWS that automatically receives updates as IBKR enhances it and fixes bugs. IBC **does not work** with the self-updating TWS, so **do not install the self-updating version for use with IBC**
- an offline or standalone TWS that, after download and installation, never changes (until you download and install another version): you **must** download and install this offline version for use with IBC.

Note that the TWS installation includes the code for both TWS and the Gateway: there is no need to do another download for the Gateway.

However, there are Gateway-specific downloads on IBKR's website. They contain the same code as the TWS downloads, but they install in a different place. You can install one of these, as well as or instead of the TWS installer. You can find these via the LOGIN dropdown in the title bar of IBKR's website.

When you run the script to load TWS, it will use the TWS installation if there is one, and if not it will use the Gateway installation if there is one. Similarly when you run the script to load the Gateway, it will use the Gateway installation if there is one, and if not it will use the TWS installation if there is one. (Needless to say, if neither a TWS download nor a Gateway download has been installed, the scripts will fail!)

It is safest to use the 'stable' offline version of TWS rather than the 'latest' version for live trading: the latter is more likely to have bugs.

IBC needs TWS to operate in English so that it can recognise the various dialogues that it interacts with. You can set TWS's language by starting it manually (ie without using IBC) and selecting the language on the initial login dialog. TWS will remember this language setting when you subsequently start it using IBC.

Note that you do not need an IBKR account to try out IBC, as you can use IBKR's Free Trial offer, for which there is a link at the top of the homepage on the website.

Where to get IBC

IBC is officially distributed as a ZIP file containing the compiled program and some additional files, detailed below. There are separate ZIP files for Windows, Linux and macOS.

The ZIP file for the latest version should be downloaded from Github. Earlier versions can also be downloaded from the same place if need be.

The distribution ZIP file contains:

- License text
- A compiled JAR (named similar to `IBC.jar`), containing the compiled Java code for the IBC program
- A sample configuration file (named similar to `config.ini`)

- Top-level script files that run IBC to start TWS or the Gateway. These files are specific to the platform (ie Windows, Linux or macOS) to which the ZIP file relates
- Windows script files that can be used to tidily shut down or restart TWS or Gateway from the same or another computer.
- A sample Windows Task Scheduler file (named similar to `Start TWS Live (daily).xml`), which can be used to automate starting TWS or Gateway on Windows systems (not present in the Linux and macOS ZIPs)
- A Scripts sub-folder containing sub-scripts used by the top-level scripts mentioned above
- A text file called `version` containing the IBC version number

Source code and build scripts are not included in the distribution ZIPs, as they are freely available from the IBC project page on Github.

Installing IBC

Installing IBC is just a matter of extracting the contents of the downloaded ZIP file to wherever you want to install it. You will make things easiest for yourself if you use the locations described in 'Default Paths' below, because that will minimise customising the configuration file and the shell scripts.

If you already have a previous IBC installation, it's wise to rename its folder (eg to `IBC.old`) so that you can easily refer back to any customisations you did for that version.

On Windows:

- create the folder where you want to install IBC, if it doesn't already exist. As noted above (see Default Paths) this is normally `C:\IBC` but it can be anywhere you like
- locate the downloaded ZIP file using File Explorer (Windows Explorer on Windows 7 and earlier). Windows treats ZIP files like an ordinary folder, so you can see its contents the same way as any other folder
- select all the files and folders and drag them into your installation folder

On Unix:

- unpack the ZIP file using a command similar to this:

```
sudo unzip ~/Downloads/IBCLinux-3.6.0.zip -d \
/opt/ibc
```

- now make sure all the script files are executable:

```
cd /opt/ibc
sudo chmod o+x *.sh */*.sh
```

Default Paths

Several script files are included in each IBC release. These script files (and these instructions) assume the default paths shown in the table below (where `<username>` represents your operating system user name, not your IBKR login id).

If you store any of these items in other locations, you will need to edit these script files to reflect this.

Platform	Item	Path
Windows	IBKR TWS program files	C:\Jts
	IBC program files	C:\IBC
	config.ini	%USERPROFILE%\Documents\IBC
Unix	IBKR TWS program files	/home/<username>/Jts
	IBC program files	/opt/ibc
	config.ini	/home/<username>/ibc
macOS	IBKR TWS program files	/home/<username>/Applications
	IBC program files	/opt/ibc
	config.ini	/home/<username>/ibc

Note that you may be able to find third-party Linux packages that allow IBC and/or TWS to be installed using a Linux package manager such as `apt`: they may not use these paths. Consult your Linux package instructions for file locations.

Password Security

To login to TWS or IB Gateway, IBC needs to know your Interactive Brokers username and password. You should very carefully secure your IBKR account username and password to prevent unauthorised use by third parties. This section gives you guidance on how to achieve this.

The username and password are given to IBC in one of two ways:

- via the configuration `.ini` file: this is the preferred method because the configuration file can be protected by the operating system
- via the command line parameters when IBC is started: this method is strongly deprecated because command line information associated with a process is easily available outside the process (for example via Task Manager on Windows)

Protecting the Configuration File

To protect this sensitive information, the configuration file needs to be stored in a location where it will not be accessible to other users of the computer. The simplest way to achieve this is to store it within your personal filestore:

- on Windows this is your `Documents` folder (which is normally actually located at:

  `C:\Users\<username>\Documents`).

  Note that this folder may also be addressed using environment variables like this:

  `%USERPROFILE%\Documents`
- on Unix it is the `/home/<username>` directory.

You are advised to place the file in its own `ibc` folder within this location.

You should also consider encrypting the folder containing the configuration file. This will prevent another user with administrator privileges gaining access to the contents: even if they use their administrator privileges to give themselves access to the file, its contents will not be decrypted because they are not the user that encrypted it.

To encrypt the folder on Windows (note that this requires a Professional or higher edition of Windows - the home edition does not provide this facility):

- right click the folder and select `Properties`
- click the `Advanced` button on the `General` tab
- set the checkbox labelled `Encrypt contents to secure data`
- finally, click the `OK` buttons to apply the changes.

Encrypting a folder on Unix is more involved, and you should refer to the documentation for your distribution.

Configuring IBC

IBC must be supplied with a configuration file. A specimen file called config.ini is included in the distribution ZIPs. You will need to edit this file to include your IBKR username and password, and to ensure that IBC behaves in the way that best suits your needs.

You should copy the supplied file from the IBC installation folder into the secure location described above before editing it, so that you have a clean copy to revert to if need be.

The sample `config.ini` file contains detailed comments on the meaning of each configuration property. Many of these have sensible defaults, or are only needed in special situations, so to help you get started quickly, here is a list of the settings that you are most likely to need to change:

Setting	Notes
IbLoginID	You must set this to your IBKR username
Password	You must set this to your IBKR password
TradingMode	You must set this to `paper` if you want to use your paper-trading account. Otherwise you can omit the setting entirely or set it to `live`.
AcceptNonBrokerageAccountWarning	Logging in to a paper-trading account results in TWS displaying a dialog asking the user to confirm that they are aware that this is not a brokerage account. Until this dialog has been accepted, TWS will not allow API connections to succeed. Setting this to 'yes' (the default) will cause IBC to automatically confirm acceptance. Setting it to 'no' will leave the dialog on display, and the user will have to deal with it manually.
IbDir	You can set this if you want TWS to store its settings in a different folder from the one it's installed in. However this usage is now deprecated because auto-restart does not work when you do this. Instead, you should specify the settings folder in

Setting	Notes
AcceptIncomingConnectionAction	the TWS_SETTINGS_PATH variable in the relevant start script. It is safest to set this to `reject` and to explicitly configure TWS to specify which IP addresses are allowed to connnect to the API, by means of the API settings in the TWS/Gateway configuration dialog.

There are two ways that IBC can locate your edited `config.ini` file.

- the simplest way is to tell it where to find the file in the script that starts IBC. In this way, you can give the configuration file any name you like. This is the recommended approach, and the supplied scripts follow this approach. If you want to change the filename from `config.ini`, or if you store it somewhere other than the default location, you'll have to edit the start script to declare its new name and location.
- if you do not specify a configuration file name, IBC will expect to find a file named `config.ini` in the current computer user's private filestore. For Windows users, the location is `%USERPROFILE%\Documents\IBC`. For Unix users, it is `~/ibc`.

Starting IBC

The normal way to start IBC is by use of a shell script. These can be identified by the `.bat` (Windows) or `.sh` (Unix) extensions. Scripts to start TWS and Gateway are included in the distribution ZIPs, and due to their complexity you are strongly advised to use them, rather than try to create your own.

Windows users can execute a shell script in a number of ways, including:

- Double-click the filename in Windows Explorer
- Create a shortcut to it on your Start menu, desktop or taskbar
- Create a scheduled task to run it automatically at the required times (see below for more information about using scheduled tasks)

If you used the default locations to install IBC and TWS, and to store your config.ini file, you should not need to edit the shell scripts. If you do need to change them, they are commented to help you.

Other Topics

Second Factor Authentication

You can use your mobile phone or tablet running Android or IOS to provide second factor authentication for your TWS login. To do this you'll need to install the IBKR Mobile app on your device, which you can download from the relevant app store. Once you've installed it, you can register it for second factor authentication via the button that it prominently displays.

Once it's registered, every time you login to TWS or Gateway (including when IBC does it for you) you'll receive an alert on your device. When you then acknowledge the alert, your login will complete.

Note that IBC cannot itself assist in the process, so you'll have to actually perform the necessary actions on your device yourself, but it's fairly convenient because you don't need to be anywhere near your computer running TWS, which is helpful if you've used some automated mechanism to start TWS.

However, if you fail to respond to the alert within a fixed period (currently 3 minutes), you will not then be able to complete your login without manual intervention at TWS, and this is where IBC *can* help. You can canfigure IBC to detect such timeouts and re-initiate the login process when this happens. To enable this behaviour you need this setting in your `config.ini` file:

```
ReloginAfterSecondFactorAuthenticationTimeout=yes
```

This timeout/relogin mechanism can repeat any number of times until you acknowledge the alert to enable login to succeed.

In some circumstances, even though you acknowledge the alert, login doesn't complete successfully. IBC can deal with this situation automatically by shutting down and restarting. This repeats the normal login sequence and thus gives you another chance to receive the second factor authentication alert on your device.

This behaviour is controlled by the `SecondFactorAuthenticationExitInterval` setting, which is the number of seconds IBC waits for login to complete when the user has acknowledged the alert, after which IBC closes down. For automatic restart, you must also set the `TWOFA_TIMEOUT_ACTION` variable in your start script file to `restart` (see the notes for this variable in the relevant start script).

If you have another automatic means of restarting IBC after it closes (for example Task Scheduler on Windows), then you should consider setting the `TWOFA_TIMEOUT_ACTION` variable in your start script to `exit`, to avoid the situation where both mechanisms react at the same time.

Scheduled Tasks (Windows only)

On Windows you can start IBC automatically using the Task Scheduler to run `StartTWS.bat` or `StartGateway.bat`.

When you define your task, make sure that the option to 'Run only when user is logged on' is selected. Doing this will ensure that you can see and interact with TWS.

You will then need to log on to Windows before the task runs.

Note that you can set up Windows to log on automatically at startup: this might be useful, for example, if your system's BIOS allows you to configure the system to power on at a particular time. Information on how to do this is freely available on the internet. But bear in mind that doing this can negatively impact your system's security.

Task Scheduler does actually allow you to specify that your task should run whether or not the user is logged in. However if you do this, the task is always started in a separate user session which you cannot see and interact with, even if you are already logged on when the task starts, or if you subsequently log on. Therefore you are strongly advised NOT to use the option for 'Run whether user is logged on or not'.

Remember also to change the task settings to prevent Windows automatically ending it after a certain time.

If you want to stop using a Scheduled Task, without losing its definition, you can right click on the task's entry in the Task Scheduler console and click 'Disable'. Clicking 'Enable' will make it available again.

You can set the AutoRestart time in the Lock and Exit section of the configuration dialog: this causes TWS/Gateway to automatically shut down and restart without requiring re-authentication at the specified time. When the restart time is reached, TWS shuts down (and IBC with it), but this does not end the task, because the `StartTWS.bat` or `StartGateway.bat` script continues running to restart IBC. The restarted IBC then reloads TWS with the relevant

information needed for it to recover its previous session without re- authentication. This sequence is then repeated each day at the same time. Thus TWS can be kept running all week, with automated startup and a single authentication at the start of the week. Note that this is all the same task, since the start script run by the Task Scheduler keeps running all the time.

Finally on the Sunday, if the task has not been ended before then, IB will prevent that session running any further because the session credentials expire. At this point it is necessary to start a new task to begin the whole cycle over again.

Since there is little point having TWS running after Friday evening (because the markets are closed), you can use the `ClosedownAt` setting in `config.ini` to tidily shut down TWS automatically after the Friday trading session has finished.

Note that TWS's auto-restart mechanism does not operate if TWS is shut down other than at the auto-restart time: for example via the File | Exit menu, or due to power failure or a program bug. This situation can be handled by configuring the task to run periodically (say every 10 minutes) during the week so that if TWS crashes or is manually shut down, the task is automatically restarted. Make sure the task is also configured to prevent a new instance if one is already running.

Note also that if you set up the task to run at user logon, and you configure your computer's BIOS to power on when power is restored after failure, and to then log on automatically, this will ensure TWS is restarted after a power outage. (Information about how to make your computer log on automatically is easily available on the internet: but make sure you understand the security implications of autologon to Windows).

IMPORTANT Make sure you use the `/INLINE` argument to `StartTWS.bat` or `StartGateway.bat` when starting IBC from Task Scheduler. Otherwise IBC starts and runs correctly, but Task Scheduler is not aware of it: in particular Task Scheduler does not show the task as running. This prevents correct operation of Task Scheduler features such as killing the task after a specified elapsed time, and periodic restarts as described above will result in multiple IBC instances being started, with unpredictable results. The reason for this is that if `/INLINE` is not used, the start scripts create a new window to run IBC in, and Task Scheduler is not aware of this, so the task ends as soon as this new window has been created.

A sample scheduled task is included in the IBC distribution ZIP, called `Start TWS (autorestart).xml`. You can import this into your Task Scheduler if you are running Windows. After importing it, you will need to enable it and change the user account it runs under.

Here is a description of how this task works. It is easiest to understand this if you first import the task and view the details in the Task Scheduler console, rather than examining the xml file.

- The task starts TWS on Sunday at 22:15 (there is nothing special about this time: choose whatever is convenient for you). As far as Task Scheduler is concerned, the task is the instantiation of the StartTWS.bat script (rather than the instantiation of IBC by the script), and when auto-restart is configured the script instantiation persists right through the various auto- restarts until TWS is shut down without auto-restart. Thus once the task is started it continues until the script ends, for example as a result of normal user exit from TWS (eg File | Exit), or a STOP command sent to IBC's command server, or a system crash.
- If there is a premature exit during the week, we would like the task to be restarted automatically. So the task specification tells Task Scheduler to restart the task every 10 minutes, but only if there isn't an instance already running.
- We don't want the task to repeat forever, so we limit the repetition to just less than 1 day (23 hours 55 mins).

- We add extra starts for Monday to Thursday, at the same time as the 'main' start (ie 22:15). This, coupled with the repetition limit, ensures that the task is restarted if it ends at any point up to 22:10 on Friday.
- We also allow the task to be started 'on demand', ie by running it manually from the Task Scheduler console.

This is not a complete description of all the task's properties, but it should be enough for you to undertand the principles behind it. There are other properties that you may want to consider using: for example, you could add another trigger to start the task as soon as the relevant user logs on.

Running with crontab (Linux only)

On Linux you can use `crontab` to run `twsstart.sh` or `gatewaystart.sh` automatically.

For example, to run `gatewaystart.sh` at 08:00 on Mondays, include a line like this in your personal crontab:

```
* 8 * * 1 export DISPLAY=:10 && /bin/bash /opt/ibc/gatewaystart.sh
```

The value you need for the DISPLAY variable will depend on how your system is configured.

Starting with IBC 3.8.1, the `twsstart.sh` and `gatewaystart.sh` scripts include a check to see if IBC is already running with the same `config.ini` file: if it is, a new instance is not started.

This enables a more sophisticated crontab entry that will periodically attempt to start IBC, but only succeed if it is not already running. For example:

```
0,15,30,45 * * * 1-5 export DISPLAY=:10 && /bin/bash /opt/ibc/gatewaystart.sh
```

will try to run gatewaystart.sh every 15 minutes from Monday to Friday. This can be useful to restart TWS/Gateway after an unexpected shutdown, or, in conjunction with the use of the `ExistingSessionDetectedAction=primaryoverride` setting in `config.ini`, to automatically restart it if using the IBKR Mobile app or the Client Portal on the IBKR Account Management page causes your TWS/Gateway session to be shut down.

Multiple IBC Instances

You may want to run more than one instance of TWS or the Gateway on the same computer, perhaps simultaneously. Here are some reasons you might want to do this:

- you want to run both your live and paper-trading IBKR accounts. This is especially true if you want to get market data from both accounts, as IBKR will only allow this if both TWS instances are on the same computer (unless you don't try to run them at the same time)
- you have multiple logins for your live IBKR account, and want to run TWS for both, perhaps at the same time
- you trade on behalf of others, perhaps your family, friends or clients, who each have their own accounts, but you want to run TWS instances for all these accounts on one powerful computer
- you want to trade in different regions at different times
- you want to test a new version of TWS in your paper trading account at the same time as using your live account in a previous version

When TWS runs, it stores a large number of settings in a folder structure (these settings may also be stored in IBKR's servers, but this may not be a useful option if you want to use multiple TWS instances). By default, TWS stores this settings folder structure in the TWS installation folder. For each username, it creates a separate folder structure. Note

however that there are some files that TWS creates while running that are not separated by username in this way, and only one instance of TWS can access them at a time. So you can run multiple TWS instances with no problem provided each instance is logged in to a different username, AND you don't try to run them at the same time.

However, by using the `TWS_SETTINGS_PATH` setting in the TWS and Gateway start scripts, you can tell TWS to store its settings whereever you like. So to have multiple IBC instances operating simultaneously, you need to create a separate start script for each instance with a different setting for `TWS_SETTINGS_PATH`. Note that you do not need to copy the TWS .jar files themselves - you can load TWS from the same installation folder for each instance.

As an alternative to having different scripts to run each instance. you could have a single script and pass the value for the `TWS_SETTINGS_PATH` variable as a parameter).

You need to ensure that the different instances don't try to write their log files to the same folder (because otherwise they might try to log to the same file, and one instance would fail).

As a concrete example, let's take the first scenario described above: you want to run both your live and paper trading accounts without them interfering with each other in any way. But before describing the steps to achieve this, here's something to bear in mind: if you have already been running either or both the live and paper TWSs, you may have already spent quite some time configuring them, and you won't want to have to repeat this work. TWS provides a means of saving and subsequently restoring settings (the `Save Settings As...` and `Settings Recovery...` commands on the `File` menu, and you can use these to keep your current settings in a temporary location, and then restore them once you've finished setting up the two instances.

So:

- install TWS into the default location (`C:\Jts` on Windows)
- create two new folders `C:\JtsLive` and `C:\JtsPaper` to store the settings
- create two IBC configuration files called `configLive.ini` and `configPaper.ini`
- set the IbDir option in them to point to the relevant folder, ie `IbDir=C:\\JtsLive` and `IbDir=C:\\JtsPaper`, and set the `IbLoginId` and `IbPassword` to the live or paper account values as appropriate
- create two start scripts (by copying `StartTWS.bat`) called `StartTWSLive.bat` and `StartTWSPaper.bat`
- change the `set CONFIG=...` line in each script file to refer to the relevant configuration file
- change the `set LOG_PATH=...` line in each script file to refer to different folders, for example `set LOG_PATH=%IBC_PATH%\LiveLogs` and `set LOG_PATH=%IBC_PATH%\PaperLogs`
- now you can run the new scripts, and each will start a separate instance of TWS connected to a different account, with its settings stored in separate folders.

Using different TWS versions simultaneously

To use more than one version of TWS (for example for testing a new version with your paper-trading account while also using a previous version for your live account), you just need to install the required versions in the normal way. Version 952 and later of TWS have installers that automatically place the relevant files in separate folders named according to the version number.

Then follow the advice in the previous section and ensure that each script file has the correct value for the `TWS_MAJOR_VRSN` variable.

How to run TWS/Gateway without IBC when IBC is installed

In order for auto-restart to work properly with IBC, the scripts that run IBC rename the TWS/Gateway executables, by appending a '1' digit to the filename (the file extension on Windows is unchanged). If these files have their original names when auto-restart occurs, then TWS/Gateway do indeed restart but they will not be running under IBC, so all the benefits of IBC will be lost. Note that for this reason you should not attempt to rename these files back to their original names while IBC is running.

The scripts do not rename the executables to their original names when IBC exits.

This causes a potential confusion if you then want to subsequently run TWS/Gateway without using IBC. Here are some suggestions:

- you can rename the excutables back to their original names before running them
- you can run TWS/Gateway directly from the renamed executables. For example in Windows you can double-click on C:\Jts\1022\tws1.exe and it will run fine
- you can edit the IB-supplied desktop shortcuts to refer to the renamed executable; or you could create additional shortcuts to the renamed executables
- if you have a script to run TWS/Gateway without IBC, you can modify the script to use either the original or renamed executable, which ever currently exists
- you could install an additional copy of TWS/Gateway into a different root folder, and only run that instance without IBC. You can use the TWS_SETTINGS_PATH variable in the IBC script to ensure that the same settings are used for both instances.

Command Server

IBC incorporates a command server that enables some aspects of its operation to be influenced by commands from external sources.

To issue a command to the command server, the command source must first establish a TCP/IP connection to the relevant port, which is specified in the `CommandServerPort` setting in `config.ini`.

The source then sends the required command (see below) as plain text, and may then read the socket for any returned data.

The source may send more than one consecutive command. When it is finished, it should send an EXIT command (though this is not necessary after a STOP command since that closes the socket automatically).

A simple way to use the command server is to make use of the `telnet` operating system command. Simple scripts are provided in the download zip for each of the commands. To use these commands, you should first edit the SendCommand.bat (for Windows) or commandsend.sh (for Unix) files to ensure the IP address and port number are correct.

The available commands are listed below. Note that none of these commands have any parameters.

STOP

Tells IBC to shut down TWS tidily, as if the user had invoked the File | Exit menu command.

RESTART

Initiates an auto-restart of TWS, as if the time specified in TWS's auto- restart setting (in the Lock and Exit section of the Global Configuration Dialog) has arrived. Note that auto-restart (and hence the RESTART command) does not require 2nd factor authentication because the credentials from the current session are re-used. Note that this command canot be used to bypass the IBKR requirement that TWS be shut down completely at some point during Sundays.

For TWS, the RESTART command is implemented by using the File | Restart… menu command, and the restart is initiated immediately.

For the Gateway, this is not possible, because it does not have this menu command. So in this case, IBC sets the auto-restart value in the Lock and Exit section of the Global Configuration Dialog: the value used is the start of the next minute if less than 58 seconds into the current minute; otherwise the start of the minute after that. Gateway also displays a transparent overlay with a countdown timer over the Gateway main window.

Note that for Gateway this auto-restart time will still be in force after the restart: to avoid further restarts at that time, you should use the `AutoRestartTime` setting in 'config.ini' to override the carried-forward time. Alternatively issue another RESTART command after restart has completed to set the auto-restart time to its usual value.

ENABLEAPI

Ensures that the 'Enable ActiveX and Socket Clients' checkbox in the API configuration is set. Note that this command is a leftover from the earliest days of IBC, and is of little (if any) use nowadays.

RECONNECTDATA

Tells TWS/Gateway to refresh all its market data connections. This is the same as the user pressing Ctrl-Alt-F.

RECONNECTACCOUNT

Tells TWS/Gateway to reconnect to the IB login server. This is the same as the user pressing Ctrl-Alt-R.

EXIT

Closes the connection to the command server.

Any Questions?

If you need assistance with running IBC, or have any queries or suggestions for improvement, you should join the IBC User Group at:

https://groups.io/g/ibcalpha

If you're convinced you've found a bug in IBC, please report it via either the IBC User Group or the GitHub Issue Tracker at:

https://github.com/IbcAlpha/IBC/issues

Please provide as much evidence as you can, especially the versions of IBC and TWS/Gateway you're using and a full description of the incorrect behaviour you're seeing. The IBC logile contains a lot of information that can often be used to rapidly diagnose the source of a problem, so attaching it to your report is always a good idea.

Changes from IBController

Although IBC has been forked from the original IBController project, it is not identical and there are several important differences that you'll need to take account of if you're switching from IBController to IBC.

Here are the main differences between IBC and IBController:

1. The program file is now called IBC.jar.
2. Changes to the settings file:
   - in IBController, the configuration settings were held in a file called `IBController.ini` by default, whereas the equivalent file in IBC is called `config.ini`
   - the setting previously called `ForceTwsApiPort` has been renamed `OverrideTwsApiPort`
   - the `AcceptIncomingConnectionAction` setting previously had a default of `accept`. This default has now changed to `manual`, which means that the user must now explicitly configure IBC to automatically accept API connections from unknown computers.
   - the settings `PasswordEncrypted` and `FIXPasswordEncrypted` have been removed, as has the facility to 'encrypt' these passwords.
   - the `IbControllerPort` setting has been renamed to `CommandServerPort`, and its default value is 0 (zero), which is taken to mean 'do not start the command server'
   - the `IbControlFrom` setting has been renamed to `ControlFrom`
   - the `IbBindAddress` setting has been renamed to `BindAddress`
3. Changes to top-level script names:

   On Windows:

   ```
   IBControllerStart.bat                -> StartTWS.bat
   IBControllerGatewayStart.bat         -> StartGateway.bat
   IBControllerStop.bat                 -> Stop.bat
   ```

   On Linux:

   ```
   IBControllerStart.sh                 -> twsstart.sh
   IBControllerGatewayStart.bat         -> gatewaystart.sh
   ```

   On macOS:

   ```
   IBControllerStart-OSX.sh             -> twsstartmacos.sh
   IBControllerGatewayStart-OSX.bat     -> gatewaystartmacos.sh
   ```