

# iBrain v4 Console Manual

---

## Activating built-in help menu

To activate the help functionality just don't provide any arguments, or type "help" as first argument.

## Configuration settings

Just like the iBrain Workbench this uses the config file to set up database access. For each database to connect to, add the following entry in the connectionStrings section, and correct it to suit your needs:

```
<add name="dbName" providerName="Oracle" connectionString="Data Source=xe;User ID=user;Password=password"/>
```

providerName can be "Oracle" or "MSSQL".

## Usage

The first argument is always the name of the connectionString entry that should be used to connect to the database.

## Import

Which is used to import from files to iBrain, the syntax is:

```
iBrainConsole [connection name] import [target] "[file list]"
```

Target describes what to import, which can be "protocoladapter" or "businessadapter".

File list is a comma separated list of full file paths which points to the files to import.

## Export

Which is used to export to files from iBrain, the syntax is:

```
iBrainConsole [connection name] export [target] [name list] [destination dir]
```

Target describes what to export. Supported types are "protocoladapter" or "businessadapter". Name list is a comma separated list of adapters to export, or just "all" to export all of the given type. Destination dir is the directory in which the exported files will be placed.

## Createpackage

Which creates a package, the syntax is:

```
iBrainConsole [connection name] createpackage [package name]
```

Package name describes the name of the created package.

## Authorizepackage

Which authorizes a created package for release, the syntax is:

```
iBrainConsole [connection name] authorizepackage [package name]
```

Package name describes the name of the package to authorize.

## Createrelease

Which creates a release, that may or may not be based on another release, the syntax is:

```
iBrainConsole [connection name] createrelease [base release] [package names]
```

Base release is the id of the release to base this release on, or "none" if it should not be based on another release.

Package names are a comma separated list of packages to include in the release.

## Demotepackage

Which demotes a package to make it available for editing.

```
iBrainConsole [connection name] demotepackage [package name]
```

Package name is a unique name identifying the package to demote.

## Addpackagefiles

Which add files to a specified package.

```
iBrainConsole [connection name] addpackagefiles [package name] "[root folder]" "[relative filepaths]"
```

Package name is a unique name identifying the package.

Root folder is the folder containing the resources to add to the package. Relative filepaths is a comma (, ) separated string of relative paths to the resources.