

## Engage Configuration

### Installer Parameters

You can pass command line parameters to the *Engage* setup executable. These parameters are helpful for unattended silent installs particularly when deploying to a larger number of users on a domain network.

```
EngageSetup.exe /qf // Default. Full UI during install.  
EngageSetup.exe /qr // Reduced UI with no wizard.  
EngageSetupe.exe /qb // Single dialog with progress bar.  
EngageSetup.exe /qn // Silent install with no UI.
```

The same command line options are available on our per-machine installs as well.

```
EngageSetupPerMachine.exe /qn
```

### Activation

Before *Engage* can be used it must be activated using a purchased product key. The product can be activated in the following ways.

#### 1. Pre-Activated – Deployed license key

This is the best option for silent installs being deployed enterprise wide. Along with the normal *Engage* installer Markido will build and provide a packaged license installer. The license installer will install a pre-activated license, configuration file and any prepared customizations. We include the company name and the expiry date of the license in the packaged filename. A sample install would use the following 2 commands:

```
EngageLicense-markido-20150521.exe  
EngageSetup.exe /qn
```

Or on a per-machine basis:

```
EngageLicensePerMachine-markido-20150521.exe  
EngageSetupPerMachine.exe /qn
```

**This is the only activation method that doesn't require internet access.**

## 2. Automatic – Command line

The option automates the activation process via the command line. This is helpful for unattended silent installs when each computer has internet access. When the installation is ran you can specify the purchased product key on the command line. The product key is then stored in a file in `%APPDATA%\Markido.Engage\` which is typically mapped to `C:\Users\ on most Windows installs. Each time Engage starts up, it checks for this product key file. If it exists and contains a valid product key Engage will try to automatically activate using the contained product key. If the activation is successful then the file is deleted. This automated activation will take place even if Engage has already been successfully activated. A sample silent install with product key command line would look like this:`

```
EngageSetup.exe /qn PRODUCTKEY="ABC12-ABC12-ABC12-ABC12-ABC12"
```

**NOTE: internet access to <https://markido.com> is required for the activation to occur.**

## 3. Manual – In app

When *Engage* is ran without an activated license it displays an activation dialog for the user to enter their purchased product key. This is a simple process of entering their Work Email and Product Key and clicking the Activate button. This will contact <https://markido.com> and activate the copy of *Engage* and then close the dialog. This can either be done by distributing the product key to employees for them to install and activate the software or by IT individuals that perform the installation and manual activation process for each computer.

## 4. Manual – via Website

If *Engage* is blocked from accessing the internet, the user can click Manual Activation... from the activation dialog once the Work Email and Product Key have been entered. This will show a second dialog that will provide a local license key to copy and paste into a web form on <https://markido.com/engage/activate>, the link is also provided in the dialog. Once the browser is open the user pastes the copied local key and clicks Activate. This will provide an activated license for the user to copy and paste back into the manual activation dialog in *Engage*. This can be done on any web browser, it doesn't necessarily need to be from the computer being installed on. Once the activated license is pasted into the textbox the user should then click Save license which will activate the copy of *Engage* and then close the dialogs.

## Runtime Configuration

Engage runtime configuration can be configured via a json config file or the windows registry.

**The json config file is recommended to avoid registry permission issues.**

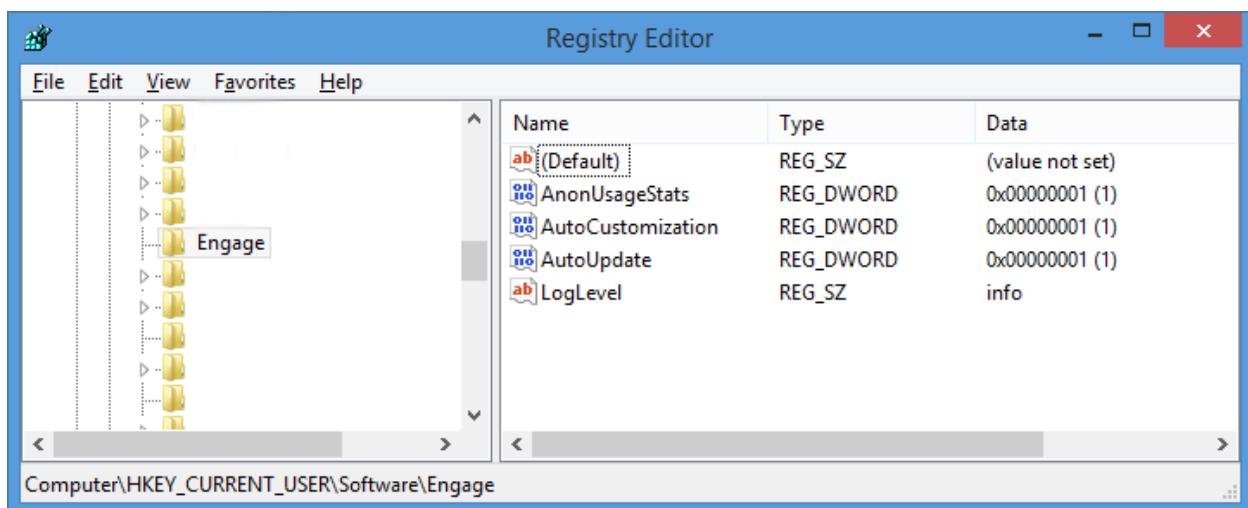
The user config file should be located at `%APPDATA%\Markido.Engage\engage.cfg` which is typically mapped to `C:\Users\<username>\AppData\Roaming\Markido.Engage\engage.cfg` on most Windows installs. A per-machine config file should be located at `%CommonAppDataFolder%\Markido\Engage\engage.cfg` which is typically mapped to `C:\ProgramData\Markido\Engage\engage.cfg`

A fully configured json file with default values would contain the following:

```
{  
  "AutoUpdate": 1,  
  "AnonUsageStats": 1,  
  "AutoCustomization": 1,  
  "LogLevel": "info",  
  "ErrorHandling": 0  
}
```

When no configuration is found the recommended default configurations are used. The only configuration setting we would suggest changing is AutoUpdate if you want to have more control over deploying updates internally. When Markido builds a pre-activated license package we include a disabled AutoUpdate and AutoCustomization configuration by default.

If you are using the registry to perform the configuration then all values are written to `HKEY_CURRENT_USER\Software\Engage`. A fully configured registry entry would look like this:



NOTE: all values must be written as the normal user of the computer and not an Administrator since the software won't be able to read it from current user later if done from the non-normal user account.

The configuration values are explained in more detail below.

### Automatic Updates (Per-User installations only)

*Engage* is by default setup to automatically update when new versions are released. To support larger computer networks the automatic update feature can be turned off. This allows builds to be tested for compatibility with user computers before a new version is deployed.

Json Config:

```
{"AutoUpdate": 0} or to turn it on: {"AutoUpdate": 1}
```

When no value is specified the default value is 1, unless it's a per-machine install in which case its disabled (0) by default.

Registry: *Engage* will read `HKEY_CURRENT_USER\Software\Engage\AutoUpdate` on start-up and will disable checking for a new version if the value is 0. If the entry is any other value or does not exist then *Engage* will check for a new version once on start-up. The following command can be used to disable this feature:

```
reg add HKCU\Software\Engage /v AutoUpdate /t REG_DWORD /d 0 /f
```

To enable the feature use the following command:

```
reg add HKCU\Software\Engage /v AutoUpdate /t REG_DWORD /d 1 /f
```

Another way to enable the feature again is to simply delete the entry for `AutoUpdate`.

**Note: Our per-machine installations do not include an auto update feature as new versions must be deployed via the domain as an administrator.**

### Anonymous Usage Statistics

*Engage* collects anonymous usage statistics. The data is anonymous and only collected for use by Markido and will never be transmitted to any third party. The data only consists of *Engage* feature usage statistics (ex. an infographic was created, an icon was inserted) that helps focus future features and releases. The collected information does not contain any data or text from the presentation.

Json Config:

```
{"AnonUsageStats": 0} or to turn it on: {"AnonUsageStats": 1}
```

When no value is specified the default value is 1.

Registry: *Engage* will read `HKEY_CURRENT_USER\Software\Engage\AnonUsageStats` on start-up and will suppress the collection if the value is 0. If the entry is any other value or does not exist then anonymous usage statistics will be collected. The following command can be used to disable this feature:

```
reg add HKCU\Software\Engage /v AnonUsageStats /t REG_DWORD /d 0 /f
```

To enable the feature use the following command:

```
reg add HKCU\Software\Engage /v AnonUsageStats /t REG_DWORD /d 1 /f
```

Another way to enable the feature again is to simply delete the entry for `AnonUsageStats`.

## Customizations

Markido provides an optional customization service to bundle provided branded assets (presentations, slides, elements, color swatch and images) into *Engage*. Once the assets are created they are provided via a deployment process. Markido hosts the customized assets on <https://markido.com> and at start-up *Engage* checks via HTTPS to see if there are assets available.

Json Config:

```
{"AutoCustomization": 0} or to turn it on: {"AutoCustomization": 1}
```

When no value is specified the default value is 1.

Registry: *Engage* will read `HKEY_CURRENT_USER\Software\Engage\AutoCustomization` on start-up and will suppress the check if the value is 0. If the entry is any other value or does not exist then auto customizations will be checked. The following command can be used to disable this feature:

```
reg add HKCU\Software\Engage /v AutoCustomization /t REG_DWORD /d 0 /f
```

To enable the feature use the following command:

```
reg add HKCU\Software\Engage /v AutoCustomization /t REG_DWORD /d 1 /f
```

Another way to enable the feature again is to simply delete the entry for `AutoCustomization`.

## Log Level

*Engage* logs certain events and errors. This is mostly for debugging purposes but can be helpful in some cases to clients. The following log levels are allowed "error", "info", "debug" and "trace". The closer to the trace level you get, the more verbose the logging is. The log is stored at `%APPDATA%\Markido.Engage\MarkidoEngage.log` which is typically mapped to `C:\Users\\AppData\Roaming\Markido.Engage\MarkidoEngage.log`.

Json Config:

```
{"LogLevel": "info"} or to keep it really quite: {"LogLevel": "error"}
```

When no value is specified the default value is “info”.

Registry: *Engage* will read `HKEY_CURRENT_USER\Software\Engage\LogLevel` on start-up. The following command can be used to change the log level:

```
reg add HKCU\Software\Engage /v LogLevel /t REG_SZ /d "info" /f
```

To keep it really quite use the following command:

```
reg add HKCU\Software\Engage /v LogLevel /t REG_SZ /d "error" /f
```

## Error Handling

If an error does occur while using *Engage*, this configuration parameter tells *Engage* how to handle it. A value of 1 will not display anything for the user but will silently send an anonymous stack trace to Markido to help us determine the cause of the error so we can address the issue if required. A value of 0 indicates to prompt the user with a dialog indicating the error and providing a choice to “Send error report” or “Don’t send” as shown in the “EngageTechnicalSpecifications.pdf”. Finally a -1 value indicates to swallow the error and don’t do anything, which is not recommended. With any of the configuration settings, the error will always get logged to the log file stored at `%APPDATA%\Markido.Engage\MarkidoEngage.log` which is typically mapped to `C:\Users\\AppData\Roaming\Markido.Engage\MarkidoEngage.log`.

Json Config:

```
{"ErrorHandling": 0} // Prompt user  
{"ErrorHandling": -1} // Swallow  
{"ErrorHandling": 1} // Silent send to Markido, default
```

When no value is specified the default value is 1.

Registry: *Engage* will read `HKEY_CURRENT_USER\Software\Engage>ErrorHandling` on start-up. The following command can be used to change the log level:

```
reg add HKCU\Software\Engage /v ErrorHandling /t REG_DWORD /d 1 /f
```

To keep it really quite use the following command:

```
reg add HKCU\Software\Engage /v ErrorHandling /t REG_DWORD /d 1 /f
```

## Uninstall

To allow you to easily script the uninstall of *Engage*, we have included an uninstall link as part of the *Engage* installation.

### Per-user:

```
start %APPDATA%\Markido\Engage\uninstall
```

### Per-machine:

```
for %A in ("c:\Program Files (x86)\Markido\Engage\uninstall") do start %~sA
```

This executes the uninstall link silently from the installation folder.

*Engage* also gets added to the Add\Remove Programs list in Windows so you can always use that to uninstall in a one off situation.