

# RepoWatchdog User Guide

[www.archimedes.com](http://www.archimedes.com)

Version 1.0.22501.2063, 2017-07-26

# Table of Contents

About .....	1
Installing.....	2
Usage .....	3
Custom emails and output files.....	4
Troubleshooting .....	6
License.....	7

# About

RepoWatchdog watches for changes to Enterprise Architect repository items and announces detected changes by email and/or writes them to a file.

# Installing

Download [RepoWatchdog installer](#) and follow the installation steps on the screen. Please note, that RepoWatchdog has following requirements for running:

- Microsoft Windows 7 or later, (32/64 bit),
- Microsoft .NET Framework 4.6 or later.



RepoDoc works with following repository types:

- MySQL
- PostgreSQL
- MS SQL Server
- Firebird (database repository)
- Oracle

For initial setup and configuration of connection to your repository, please follow the Sparx Systems Help.

Once you have finished the installation you can start RepoWatchdog from the command line.

# Usage

RepoWatchdog watches for changes to Enterprise Architect repository items and announces detected changes by an email and/or writes them to a file. It works in one of two modes:

1. in a **periodic** mode, the watchdog checks the repository continuously for changes with a specific period,
2. in a **batch** mode, the watchdog checks the repository once and looks for changes between specific dates.



RepoWatchdog works by analyzing the audit log and for this reason the audit log must be enabled in the project.

Following are the examples how to use RepoWatchdog. Parameters between `<` and `>` have to be substituted with real values.

*Periodic check of the repository (each 10 seconds). Each detected change is sent by email (default mail template is used).*

```
RepoWatchdog.exe -c DSN=<dsn> -s <smtp_server> --username <username> --password  
<password> --from <sender_email> --to <recipient_email>
```

*Periodic check of the repository (each 60 seconds) with more detailed console output. Each detected change is sent by email (default mail template is used).*

```
RepoWatchdog.exe -c DSN=<dsn> -s <smtp_server> --username <username> --password  
<password> --from <sender_email> --to <recipient_email> -l info -p 60
```

*One time check of the repository. All changes from May 2017 are written to a HTML file.*

```
RepoWatchdog.exe -c DSN=<dsn> -b 20170501-20170531 -t "c:\Program Files  
(x86)\Archimedes\RepoWatchdog\Templates\File.liquid"
```



Be careful when using batch mode with the default mail template. It may cause to send too many mails as each change is sent in a separate mail. Use the file template in these cases or create your own mail template to announce all changes in a single email.

# Custom emails and output files

The content of emails and files produced by RepoWatchdog is determined by templates and RepoWatchdog comes with two pre-installed templates for sending changes by emails or writing them to a file. These default templates may be found in the installation directory which typically is `c:\Program Files (x86)\Archimedes\RepoWatchdog\Templates\`. You are free to copy and modify these templates to achieve different formats of emails and output files. By modifying the templates you can group changes together or send one email will all changes.

RepoWatchdog doesn't use own templating language, but instead relies on [Liquid template language](#). Please have a look on the [introduction to Liquid](#) if you are not familiar with it.

Beside the standard Liquid objects, tags and filters, RepoWatchdog introduces these additional objects and filters:

- Objects:
  - `dsn` contains the user supplied data source name argument.
  - `mode` contains either the value `periodic` or `batch` depending on the mode in which RepoWatchdog runs.
  - `start_date` and `end_date` contain the user given dates when the watchdog runs in `batch` mode and are empty otherwise.
  - `changes` is an array of objects containing information about the changes that have been found. Each array item contains following information:
    - `type` contains the type of the change: `INSERT`, `UPDATE` or `DELETE`.
    - `item_name` contains the repository item name that has been changed. This is typically the name found in the Project browser in Enterprise Architect.
    - `item_guid` contains the GUID of the repository item that has been changed.
    - `item_type` contains the type of the repository item that has been changed. Possible values include `Package`, `Class`, `Actor`, `Component` ... See the [Element class attributes](#) and look for Type attribute for all values.
    - `user_name` contains the user name or login of the user that made the change.
    - `properties` is an array of objects containing information about the changed properties. Each array item contains following information:
      - `name` of the property, like `Name`, `Alias`, `Notes` ...
      - `old_value` contains the old (original) value.
      - `new_value` contains the new (modified) value.
- Filters:
  - `send_email` filter sends an email using the smtp settings provided as input arguments. The subject and body have to be provided in the template as objects:

```
{{ mail_body | send_email: mail_subject }}
```

- `write_file` filter writes to a file. The content and path have to be provided in the template as objects:

```
{{ file_content | write_file: file_path }}
```

# Troubleshooting

RepoWatchdog outputs information to the console and log file depending on the log level. The log level is determined by the config file in the installation directory, but may also be set using commandline argument to one of the following values: *error*, *warn*, *info* and *debug*. The config file uses [log4net](#) format.



Use the *info* or *debug* levels to troubleshoot and use the default (*warn*) level for normal operations.



# License

RepoWatchdog is able to look for changes in following repository items: models, packages, elements, diagrams, attributes, operations, operation parameters.

Without a valid license key, the RepoWatchdog works only with models, packages, elements and diagrams. Time limited trial license key is available for free and can be simply obtained by running the command bellow.

*License key download command.*

```
RepoWatchdog.exe --downloadlicensekey  
  
09:25:38 WARN - Downloading license key ...  
09:25:49 WARN - License key downloaded.
```

To show the license key validity period run command below.

*License key information command.*

```
RepoWatchdog.exe --showlicensekey  
  
09:26:18 WARN - License key information:  
09:26:18 WARN - License key is valid until 30. 6. 2017.
```

During the validity period, RepoWatchdog supports all repository items mentioned above.

To obtain a registered license key run command below.

*Computer ID information command.*

```
RepoWatchdog.exe --showcomputerid  
  
09:26:40 WARN - Computer ID:  
6732659e5ea4bddb3a518fbe6fe5148814f075025cc7a40cf8518be284f45661
```

and contact [info@archimetes.com](mailto:info@archimetes.com) with your computer ID attached.

To install a registered license key, set the customer ID first with the value provided by Archimetes representative.

*Setting customer ID information command.*

```
RepoWatchdog.exe --setcustomerid your_customer_id  
  
09:45:13 WARN - Setting customer ID to: your_customer_id
```

and then issue the download license key command again.