

Upgrade Guide

1 Verify System Requirements

Before starting with the upgrade, please verify that you have the required software packages installed and upgraded to the required version. See here for details: [Installation Prerequisites](#)

2 Preparation

This guide provides instructions to upgrade iteraplan from release 5.5 to release 6.0/6.0.5. Please note that this guide is only applicable to these versions of iteraplan. For prior versions, please follow the appropriate iteraplan upgrade guides in order to upgrade to iteraplan 5.5 first.

The upgrade steps only need to be carried out once. However, they must be repeated for every installed instance of iteraplan (if applicable). The upgrade process is not intended to be reversible, i.e. after performing the upgrade there is no standard way to revert to the previously installed version of iteraplan. For that reason, we **strongly recommend to create a complete backup** of your entire iteraplan database and the application files.

To help you carry out the upgrade our [checklist](#) might prove useful.

2.1 Conventions and Assumptions

In the scope of this guide, it is assumed that iteraplan is deployed on an Apache Tomcat web server. The Tomcat installation directory is hereafter referred to as `$TOMCAT_HOME`, regardless of the underlying operating system. Furthermore, it is assumed that the iteraplan application is deployed under `$TOMCAT_HOME/webapps/iteraplan`. If you have deployed the application under a different name, replace `iteraplan` in the provided path as appropriate.

The Java Runtime Environment (JRE) or Java Development Kit (JDK) installation directory is denoted as `$JAVA_HOME`.

This guide uses Unix conventions for path names. For Windows, replace slashes (/) with back-slashes (\) and include drive letters where necessary.

3 Upgrade

3.1 Upgrade the database structure

Be sure that you have shut down iteraplan or the entire Tomcat instance. To shut down Tomcat on Windows, use the Computer Management Console to stop the Tomcat Service. Alternatively, run the batch script `shutdown.bat` in the `$TOMCAT_HOME/bin` directory. On Unix-like systems, use the normal daemon control mechanism, i.e. `/etc/init.d/tomcat stop`, or run the `shutdown.sh` shell script in the `$TOMCAT_HOME/bin` directory.

All database upgrade scripts are encoded with UTF-8. Please ensure that you use this encoding.

Depending on whether you use MySQL, Oracle or SQLServer, use the SQL script for the respective database vendor. The script will perform all necessary modifications of your database.

In the directory `upgrade/v5.5To6.0/` three files are provided:

```
migration_iteraplan_550_to_60_[database].sql
```

For the upgrade to release 6.0.5 the directory `/v6.0To6.1/` contains three more files:

```
migration_iteraplan_600_to_605_[database].sql
```

Replace the placeholder `[database]` with the database you use. To execute the scripts, use a database management tool appropriate for your system. Because of the wide variety of tools, detailed instructions are omitted in this guide.

3.2 Migrate history data

If

- the history function has never been active in the past or
- you already migrated the history data when upgrading to the release 5.4 / 5.5 or
- you do not want to keep the history data,

skip this chapter completely and proceed with chapter 3.3.

If you want to migrate your history data from previous version, please proceed with the following steps.

CAUTION

The history migration will take several hours. Our internal test show times from **5 up to 20 and more hours**. We strongly recommend to run this long-running job **over night and/or on a weekend**. During this job iteraplan needs to be shut down to prevent data inconsistency.

This migration needs to be run only once.

iteraplan 5.4 and later comes with a reworked implementation of its historization functionality. If the history functionality has been enabled in older iteraplan versions and you want to migrate that history data to be used in iteraplan 6.0, the provided history migration tool has to be executed once.

To run the history migration tool, proceed as follows:

1. Backup your iteraplan database, if not already done.
2. Extract the provided `historyMigrationTool.zip` file into a folder. The folder has to be located on the same computer or virtual machine like iteraplan is running on.

After extraction of the zip-file, the folder should contain the following files:

- `historyMigrationTool.jar`
 - `migrateHistory.bat` (for Windows systems)
 - `migrateHistory.sh` (for Unix systems)
3. Copy the following files from your old iteraplan installation to the folder with the migration tool:
 - a. from the `$TOMCAT_HOME/webapps/iteraplan/WEB-INF/classes` folder: `iteraplan.properties`, `iteraplan-db.properties` and `iteraplan_local.properties`
 - b. from `$TOMCAT_HOME/lib`: The jdbc driver jar file corresponding to your database:
 - for MS SQL Server: `sqljdbc[version].jar` (for example `sqljdbc4.jar`)
 - for Oracle: `ojdbc[version].jar` (for example `ojdbc6.jar`)
 - for MySQL: `mysql-connector-java-[version].jar` (for example `mysql-connector-java-5.1.31.jar`)

If more than one of those files exists, **only** copy the file for the database you use with iteraplan.

4. Make sure the environment variable `JAVA_HOME` is set correctly.
5. Run the shell script appropriate for your operating system: `migrateHistory.sh` on Linux systems or `migrateHistory.bat` on Windows systems.

During the migration the history migration tool uses two different log-files. In the file "migration.log" the progress of the tool is tracked. It contains one entry for each element that is migrated. The file "iteraplan.log" contains eventual exceptions.

If it is not possible to migrate a single change set, this is logged in both files. The migration of other changesets is not affected in this case. If there is an error during the migration, please contact the iteraplan customer support.

3.3 Configure the new installation

Before iteraplan 6.0 is set up, the previous installation of iteraplan should be completely undeployed from the Tomcat instance. In particular, the following files and directories should be deleted from the Tomcat installation directory:

```
$TOMCAT_HOME/conf/Catalina/localhost/iteraplan.xml
$TOMCAT_HOME/webapps/iteraplan.war
$TOMCAT_HOME/webapps/iteraplan
```

If your old iteraplan installation is deployed simultaneously with iteraplan 6.0, unexpected side effects may occur due to competing database and file access.

Launch iteraplan installer

Run the installer by launching the `iteraplanInstaller.jar` file and follow the instructions. Depending on your system configuration, double-clicking the jar file might not work. If this is the case, the file can be launched from the command line by executing

```
java jar iteraplanInstaller.jar
```

or by explicitly stating the JRE to use:

```
$JAVA_HOME/bin/java jar iteraplanInstaller.jar
```

Enter database connection parameters

During the installation process, the installer will request you to provide database connection parameters. The parameters to use are the same as the ones used for the installation of previous versions of iteraplan.

If you are not sure, you can look up the settings in the following two files of your old version's backup:

`$TOMCATBACKUP/webapps/iteraplan/WEBINF/classes/iteraplantdb.properties` for the iteraplan database and `$TOMCATBACKUP/conf/Catalina/localhost/iteraplan.xml` for the iTURM database with authentication information.

Important: Make sure to disable database initialization. Otherwise, your current database contents will be deleted completely and you will need to resort to the data you have backed up.

Enter full-text search index storage directory

In a subsequent step, you will be requested to specify a directory where the files of the search index are to be stored. It is recommended to use a new empty subdirectory of the Tomcat installation directory, for example `$TOMCAT_HOME/indexes`.

In case several instances of iteraplan are deployed with different database configurations, it is important to use different search index directories.

Empty Tomcat's temporary work directory

Tomcat's temporary work directory probably contains files of a previously installed version of iteraplan, even after installing the new version. This might cause problems. Therefore delete the following folder:

```
$TOMCAT_HOME/work/Catalina/localhost/[iteraplan]
```

This forces Tomcat to recompile at start up.

Deployment on Tomcat and fine-tuning the configuration

After iteraplan 6.0 has been successfully configured by the installer, the generated WAR-file should be automatically copied into the Tomcat webapps directory (`$TOMCAT_HOME/webapps`). Then you can start Tomcat in order to deploy iteraplan 6.0.

During deployment, a directory `$TOMCAT_HOME/webapps/iteraplan` (corresponding to the name of the WAR-file), in which the iteraplan installation resides, will be generated. In case you made any customizations to your old installation's configuration options, make sure to transfer the modifications analogously, for example to the the properties in `$TOMCAT_HOME/webapps/iteraplan/WEB-INF/classes/iteraplan.properties` file or to any other *.properties file in `$TOMCAT_HOME/webapps/iteraplan/WEB-INF/classes`

Copy document templates

iteraplan allows to store document templates for some report types on the server. Copy those template files from your old installation's backup to the new iteraplan 6.0 installation. To do that, go through all sub-directories in `$TOMCAT_BACKUP/webapps/iteraplan/WEB-INF/classes/templates` and copy all files from the old installation which do not yet exist in the new installation's corresponding directory. That is, you should not overwrite existing files, e.g.

```
.../templates/elasticExcel/ExcelWorkbook.xlsx.
```

It's okay if no files are to be copied.

Copy Graphics Reactor Contents

The iteraplan Graphics Reactor stores all input and output files in the file system. Copy the contents of `$TOMCAT_BACKUP/webapps/iteraplan/WEB-INF/classes/reactor` from the backup to the corresponding directory in the new installation. It's okay if no files are to be copied.

Copy Plugin API Scripts

The iteraplan Plugin API scripts are stored in the file system. Copy the contents of `$TOMCAT_BACKUP/webapps/iteraplan/WEB-INF/classes/scripts` from the backup to the corresponding directory in the new installation. It's okay if no files are to be copied.

Enable Compression within Tomcat

To speed up the initial page load of the new iteraplan client (amongst other operations) we recommend to enable compression in the Tomcat connector. This is not mandatory, but generally a good idea.

If you want to enable compression, add the following two settings to the connector entry in the Tomcat `server.xml` file:

```
compression="on"
compressableMimeType="application/json,text/html,text/xml,
text/plain,text/css,text/javascript,application/javascript"
```

Your connector entry might then look like this:

```
<Connector executor="iteraplanThreadPool"

    port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443"
    compression="on"
    compressableMimeType="application/json,text/html,text/xml, text/plain,text/css,text/javascript,application
/javascript"
/>
```

Transfer LDAP authentication settings

If iteraplan was installed with LDAP authentication enabled, simply copy the `iteraplan-auth.properties` file from your old installation's backup to the `$TOMCAT_HOME/webapps/iteraplan/WEB-INF/classes/` directory. The contents of the file do not need to be adjusted.

Final restart

As a final step, restart Tomcat to make sure that any modifications to the iteraplan configuration become active.

3.4 Notes

After the upgrade, some users might report that iteraplan does no longer work as expected. Possible observations might include that buttons have wrong titles or that the client freezes when performing certain tasks.

In this case users should clear their local browser cache and re-login into iteraplan.