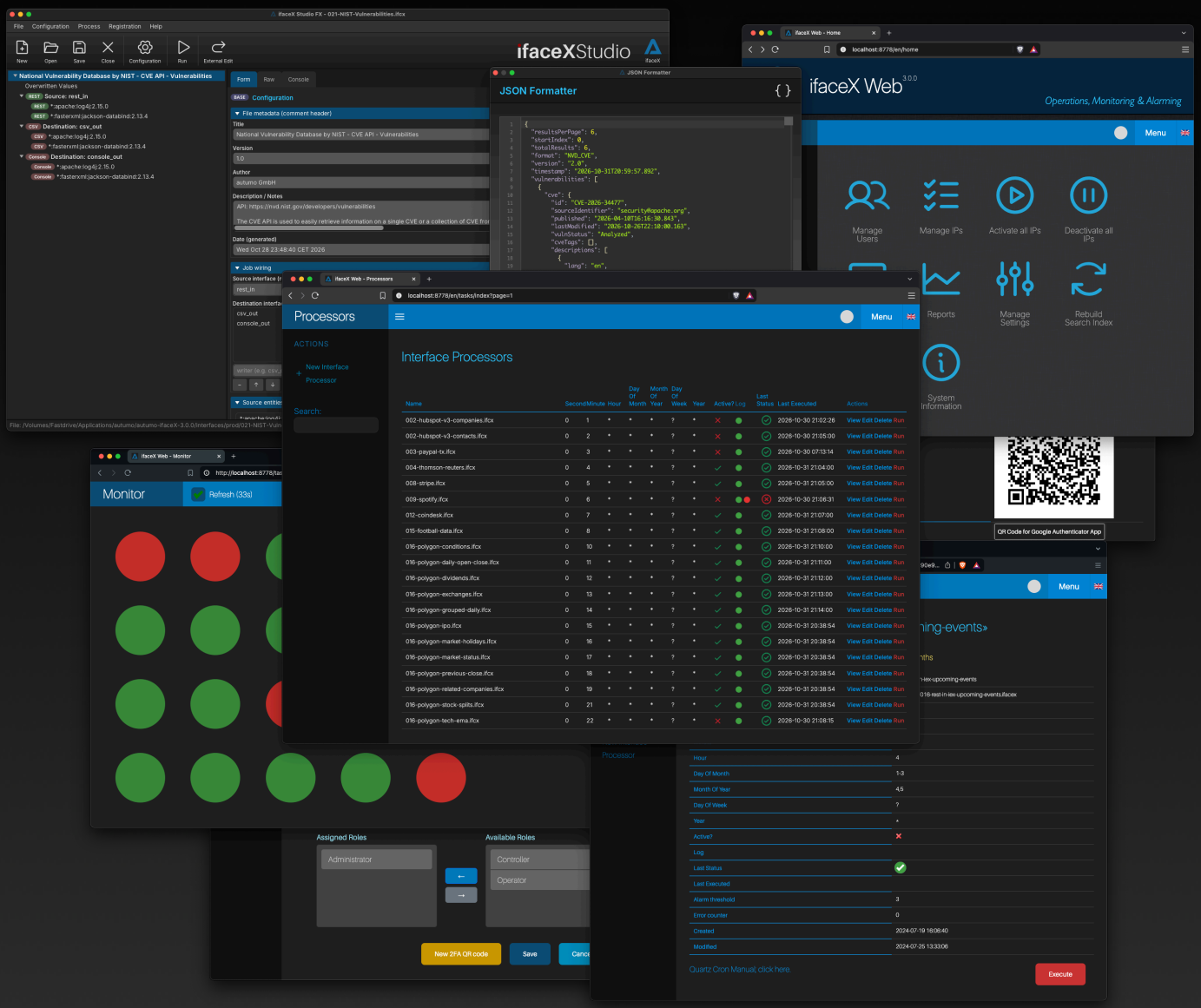
Quickstart Guide

ifaceX



© 2026 autumo GmbH · Switzerland

Legal information

This manual may not be copied, reproduced, passed on or published without the author's consent.

© 2026 autumo GmbH

Table Of Contents

1. Lightspeed 1
2. Introduction 2
3. Preconditions 3
4. Basic Installation 6
5. Deployment & Configuration 8
   5.1. Database Setup 8
   5.2. ifaceX Web 10
6. Startup 12
7. Final Remarks 15

List Of Figures

Figure 1: Basic Deployment **7**
Figure 2: Deployment with ifaceX Web **10**
Figure 3: ifaceX Server UNIX Console Start **12**
Figure 4: ifaceX Web Default View **14**
Figure 5: ifaceX Web Monitor **15**

© 2024 autumo GmbH



1. Lightspeed

Without any knowledge, the quickest Quickstart is:

1. Extract „`autumo-ifaceX-x.y.z-nn.zip`“ or „`autumo-ifaceX-x.y.z-trial.zip`“ (demo version) anywhere.
2. Start the ifaceX server with the command „`bin/server.sh|bat start`“.
3. Browse to: http://localhost:8778
4. User: `admin`
   Password: `ifacex`

Notes:

- The non-trial version must be registered, otherwise the server will not start. The demo/trial version doesn't need to be registered!
- It is assumed that Java 11+ (JRE or JDK) is installed. Note that you can quickly install an Azul Zulu build of OpenJDK with the. „`bin/install_java.sh|bat`“ script.



2. Introduction

This „**Quickstart Guide**" shows the shortest path to run the autumo ifaceX platform and all of its modules with minimum configuration. It doesn't replace the „**ifaceX Manual**" that comes with the purchased product. It contains in-depth configurations and deployment scenarios; ifaceX is highly customizable and configurable, even the shipped default configuration works for most environments. If you have acquired an Enterprise Edition License, you also get a „**Developer Reference**" manual for extended customization.

ifaceX is a secure and scalable enterprise interface exchange and monitoring platform that transfers data from one endpoint to another, such as REST APIs, databases, CSV files & mails with interfaces like HubSpot, PayPal, AWS S3, Spotify or financial APIs, etc. Moreover, it simply executes any process or even programming code that can be monitored as well. Processing can be customized all-including flexible target field mapping based on formulas and data mapping through a drag'n'drop user interface. ifaceX consists of the following modules:

- **ifaceX**:
  - Run your Interface Processors (IPs); all at once, as many as you want or a single IP
  - Execute the IPs in different modes (atomic/serially/parallely) & configure processing
  - Besides sending data to REST, databases, files and mails etc., you can run and any system program
- **ifaceX Scheduler**:
  - Run ifaceX as a server that executes IPs with an internal scheduler (Quartz Scheduler)
- **ifaceX Web**:
  - Schedule and surveil your Interface Processors in detail with the scheduler
  - Access Interface Processor data (e.g. status) per JSON REST
  - Operation monitoring and alarming with threshold levels (Email & SMS)
  - Available roles: Administrators, Operators & Controllers
- **ifaceX Studio**:
  - Configure | Develop your IPCs within a GUI
  - Configure your ifaceX installation and test your IPCs
- **ifaceX Console Tools**:
  - Run ifaceX with your OS scheduler and console dev-tools
  - Protect all sensitive data with advanced cryptographic algorithms
- **ifaceX Templates**:
  - Provides lots of Interface Processor Configuration (IPC) templates to connect to many well-known cloud services per REST API out-of-the-box, e.g., PayPal, IEX Cloud, SWIFT, Hubspot, Spotify, etc. (continuously supplemented)



3. Preconditions

1. You have downloaded autumo ifaceX and you are in possession of a file named „`autumo-ifaceX-x.y.z-nn.zip`" or „`autumo-ifaceX-x.y.z-trial.zip`" (demo version).

   nn = amount of interface processor runnable per execution depending on your license.

   The demo version can execute 5 interface processors and can be tried out for 30 days.

   Note: NEVER EVER execute autumo ifaceX or the ifaceX Studio when it hasn't been downloaded from https://products.autumo.ch, https://autumo.ch, https://github.com/autumoswitzerland/autumo-ifacex-studio-fx-public or it hasn't delivered personally to you.

   Generally, purchased autumo enterprise products like ifaceX are only available within the customer-protected area.

2. You want to operate autumo ifaceX on Linux-, UNIX- or Windows-based systems.

3. If you have purchased a copy you have your license ready for registration, which can be done with the command line tool „`register.sh|bat`" or with the free ifaceX Studio that can be downloaded from https://github.com/autumoswitzerland/autumo-ifacex-studio-fx-public.

   Note: The registering process needs to call the autumo license server on port 443 once during the registration process. The demo version doesn't need to be registered!

4. autumo ifaceX runs on the Java platform which is not delivered within the package. You have to procure & install Java by your own. You need to have a Java runtime implementation based at least on the version 21 specification: https://jcp.org/en/jsr/detail?id=396.

   autumo ifaceX is tested on Azul Zulu Builds of OpenJDK, Oracle Builds of OpenJDK as well as on the Oracle Java SE. We suggest you use the following Java runtime:

   - Azul Zulu Builds of OpenJDK 21+
     https://www.azul.com/downloads
   - Oracle's Build of OpenJDK 21+
     https://openjdk.org/projects/jdk



- Oracle Java SE 21+
  https://www.oracle.com/java/technologies/javase-downloads.html
- Builds of OpenJDK 21+ from other vendors (untested); check Foojay's JDK Almanac to get an overview of OpenJDK Builds by various vendors:
  https://foojay.io/download /
  https://foojay.io/almanac/jdk-21

If you are using Oracle's Java SE platform, you have to obtain a license from Oracle. Any Java version above 21 and even other reference implementations by other vendors (e.g., Eclipse Temurin, Bellsoft Liberica JDK, etc.) run autumo ifaceX, but make sure to test your Java runtime with ifaceX first: A demo version is available at https://products.autumo.ch.

Azul Zulu Builds of OpenJDK

All of autumo's enterprise products run on Azul Zulu Builds of OpenJDK, and the GUIs even ship with their OpenJDK implementation. Azul is one of the vendors that even closes critical security gaps to older JSR specifications and their builds are all TCK tested. If you are in the situation of having to choose a JRE/JDK, we recommend you take a look at the Zulu community. If you ever find yourself in a situation where you need to improve the performance of your environment, Azul provides builds with support and even high performance builds. Azul is an official partner of autumo GmbH.

You can easily install an Azul Zulu build of OpenJDK with the. „`bin/install_java.sh|bat`" script, which the is only used for ifaceX.

5. autumo ifaceX comes with the following database drivers:

   - H2 - https://www.h2database.com/html/main.html
   - MariaDB - https://mariadb.org (Works for MySQL too!)
   - PostgreSQL - https://www.postgresql.org
   - MongoDB - https://mongodb.github.io/mongo-java-driver
     [used by Interface Processors that connect to a MongoDB]

6. If you want to connect to an Oracle database you have to download Oracle's JDBC database driver libraries that can be found here:
   https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html.

   If you use Oracle's JDBC database driver libraries, you have to obtain a license from Oracle. After you have downloaded them, place them into the autumo ifaceX „`lib/`" directory; they will be loaded automatically when using the „`run.sh|.bat`" script.



7. **We suggest the following version**:
   Oracle Database 26ai (23.26.4.0) JDBC Driver & UCP Downloads (Long Term Release). Package: `ojdbc17.jar` - Implements JDBC 4.3 spec and certified with JDK17, JDK19, JDK21 and JDK25.

   A word when using MySQL: Due to the GPL license, we don't distribute or create a dependency to the MySQL Connector for Java. Visit Oracle MySQL website and download it yourself if you want to use this connector. Note that MariaDB Connector/J also supports MySQL databases, including MySQL 5.5 and later versions. Also have a look here for further valuable information in this context: https://mariadb.com/kb/en/licensing-faq.

8. Precondition when using system schedulers for execution: Your operating system contains a scheduler that can execute batch files on Windows and Shell-scripts files on UNIX or Linux.



4. Basic Installation

Note: The mentioned directories are only suggestions. IT operations usually has its own structure that should be followed.

Steps:

1. Extract your ZIP-package to the target directory on target platform or somewhere else, if you want to register your software on another platform. The application needs read- as well as write-access to this directory.

2. On UNIX and Linux we suggest you place your software on the target platform within the „`opt/`" directory, e.g., „`/home/<app-user>/opt/autumo-ifaceX-x.y.z`". We also suggest you create a symbolic link „`/home/<app-user>/opt/autumo-ifaceX`" that targets the mentioned directory path containing the current version. If you get major updates you simply re-link to the new version.

   Note: ifaceX needs read and write access within its own user directory!

   We further suggest, that you copy the directory „`interfaces/`" somewhere else if not already done in a previous version. This directory contains all IPCs that should be treated as source files and hence, the original files should be administrated by an in-house source versioning system. These IPCs can be created manually or with the ifaceX Studio. After moving this directory elsewhere, you should create a symbolic link that points to the new location where IPCs now are stored that preserve the original „`interfaces/`"-directory structure. This might be not necessary if your deployment process copies the newest IPC-set always to the correct autumo ifaceX directory.

   Note: Generally, you are free to define any directory for IPCs. All modules support relative as well as absolute paths!

   Likewise on Windows, you place autumo ifaceX within the directory „`C:\Program Files\autumo-ifaceX`" and create comparable symbolic links on your Windows target platform if this approach suits.

Such an installation runs configured single or multiple IPs, serially or parallely. You simply execute the „`run.sh|.bat`" script.

This basic deployment allows you to run Interface Processors on the console manually or by using a OS scheduler such as UNIX cron-jobs or with Window's own scheduler.

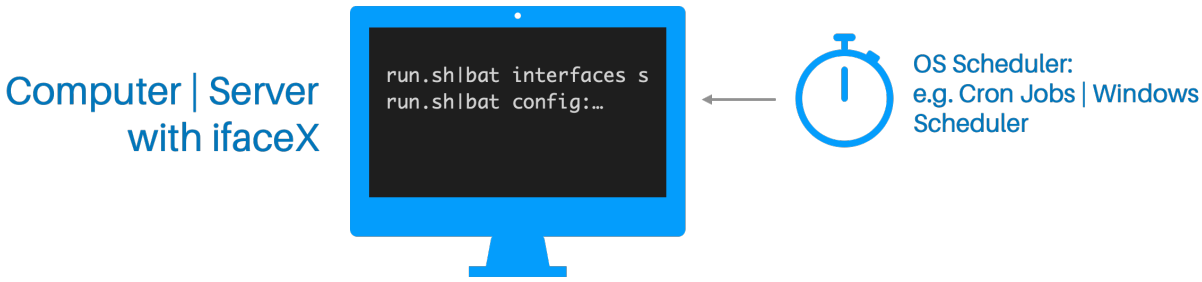


Figure 1: Basic Deployment

In this case you don't need a database at all, just a computer or server where ifaceX is deployed.

The setup of ifaceX Web and its modules with a database is described in the next chapter.



5. Deployment & Configuration

The ifaceX platform can be deployed in many ways and give you the freedom to suit your operation environment — small operation environments such as hosted services or a big IT operation center. The following instructions show how you deploy and configure ifaceX Web with its scheduler module as a standalone-server. The manual also describes the setup of the server in detail and in a distributed scenario where the server is separated from the ifaceX web application, which then runs in a web container such as Apache Tomcat or Oracle WebLogic.

5.1. Database Setup

If you want to use the internally shipped H2 database (https://www.h2database.com), you have to do **nothing**, except backup it regularly (e.g., daily). This is usually a small company scenario. It is pre-setup and pre-configured ready to use and to run. It is located within the directory „`db/h2/db`" and the default password is „`ifacex`".

Note: In any case, at this point you can already start the ifaceX server with the command „`bin/server.sh|bat start`" without performing any further actions!

If you would like to setup another database, follow the steps below. Note that the setup of the H2 database is also explained in case you want to set it up yourself.

ifaceX officially supports and is tested with the following databases:

- H2 Database Engine: https://www.h2database.com
- MariaDB: https://mariadb.org
- MySQL: https://www.mysql.com
- PostgreSQL: https://www.postgresql.org
- Oracle: https://www.oracle.com/database

If your environment is a typical outsourced hosting service, then you quite likely are able to use MySQL or MariaDB; today it is often MariaDB for good reasons. If you are a freak and you like PostgreSQL, which is a fantastic and easy to use database, you setup PostgreSQL.

Depending on your chosen database, you have to install the following SQL scripts:

1. Scripts for the Quartz Scheduler:

- H2 Database Engine: `db/install_quartz/qrtz_tables_h2.sql`
- MariaDB & MySQL: `db/install_quartz/qrtz_tables_mysql.sql`
- PostgreSQL: `db/install_quartz/qrtz_tables_postgres.sql`
- Oracle: `db/install_quartz/qrtz_tables_oracle.sql`



2. Scripts for the ifaceX Server (ifaceX Web + Scheduler)

- H2 Database Engine: `db/install_ifacex_h2.sql`
- MariaDB & MySQL: `db/install_ifacex_mysql.sql`
- PostgreSQL: `db/install_ifacex_postgres.sql`
- Oracle: `db/install_ifacex_oracle.sql`

Adjust the following parameters in „`cfg/ifacex-server.cfg`“:

```
#
# DB URL.
# Note: for servlet context,
#       define absolute path
#
db_url=jdbc:h2:./db/h2/db/ifacex;AUTO_SERVER=TRUE;IFEXISTS=TRUE
#db_url=jdbc:mariadb://localhost:3306/ifacex
#db_url=jdbc:postgresql://localhost:5432/ifacex
#db_url=jdbc:oracle:thin:@localhost:1521/orcl

#
# DB User.
#
db_user=ifacex

#
# DB password.
#
# It is encoded or not, see switch 'admin_pw_encoded'.
#
# If it is encoded, the ifaceX encoder algorithm 1 is
# used; it doesn't need to be encapsulated in curly
# brackets nor a mode '1:..' must be specified.
# Only the encoded password after the colon ':'
# shown in 'encoder.sh' output is used here.
# Use ‚encoder.sh‘, use mode 1.
#
db_password=ifacex
```

There are more DB parameters, but the above are enough for a minimal setup. Just use the pre-configured DB URL's and uncomment (remove „#“) the right line and adjust them according to their Java JDBC specification. **Make sure only one DB URL is uncommented**! Also, use your user and password for the database.

Your minimal database setup is fully done!

5.2. ifaceX Web

The following deployment allows you to run Interface Processors according to a very specific time plan or periodically without external or a system scheduler and allows you to monitor professionally all Interface Processors as well having a state-of-the art alarming mechanism for surveilled IT operations. For this, the ifaceX Server is deployed too.

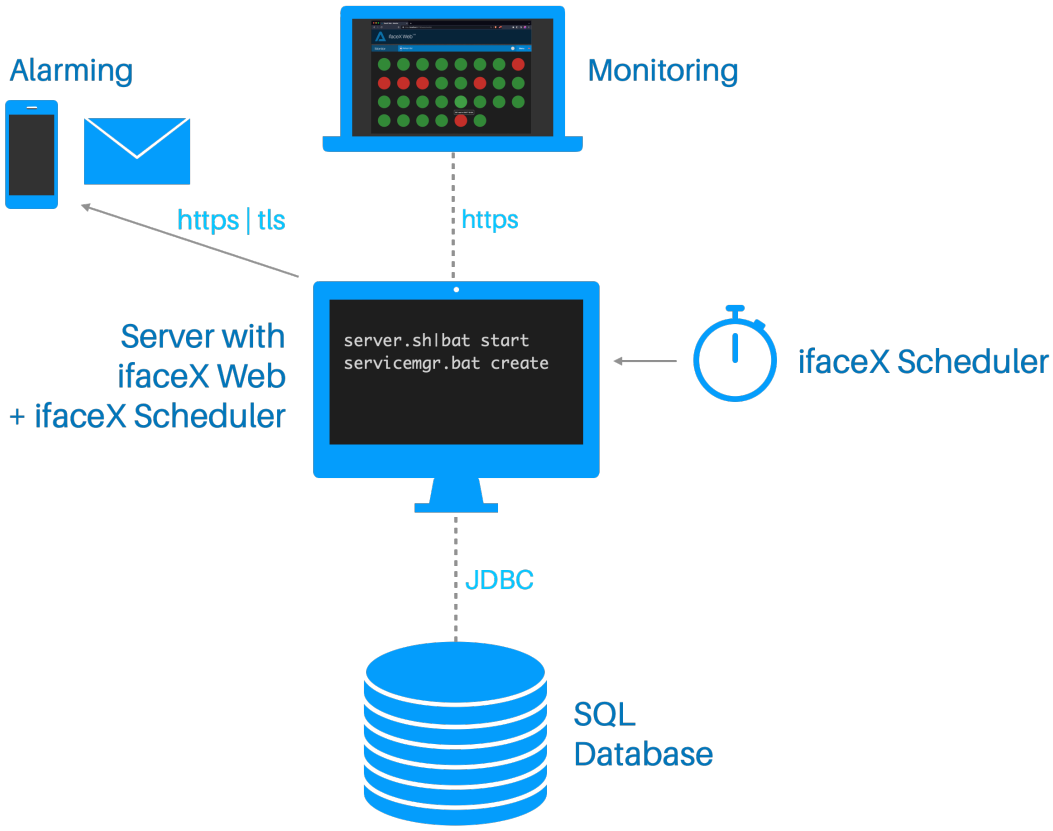


Figure 2: Deployment with ifaceX Web

This deployment is also called „stand-alone" deployment, because the ifaceX Server contains a web-server for the monitoring and no additional web-container such as e.g. „Tomcat" must be setup. It is a very comfortable way to use this deployment scenario.

Note: Further deployment scenarios and the use of different and secure communication protocols between the ifaceX Server and ifaceX Web, including HTTP/HTTPS-tunneling, is described in the „**ifaceX Manual**" — the ifaceX platform is highly configurable.



Adjust the following parameters in „`cfg/ifacex-server.cfg`“:

```
#
# Change this ONCE! But this will make all
# earlier generated passwords invalid, configure
# this from beginning!
#
# NOTE:
# - It must be a HEX string of 32 character length.
# - it must be the same within 'ifacex.cfg' and
#   the one used in the 'ifaceX Studio'!
# - If you run ifaceX Web distributed; then the
#   'ifacex-server.cfg' within the servlet container
#   must have the same seed too!
#
# You can use this generator:
# https://www.browserling.com/tools/random-hex
#
secret_key_seed=ee90f44e4db7cec40e5ec86c5097cad7

#
# Web address, URL.
# As seen from the outside world!
# Only used for stand-alone version.
#
# E.g. https://mypage.com
#
ws_url=http://localhost
```

Also, if you want alarming and password reset mails to work, setup all necessary mail-parameters beginning with „`mail_`“; they also have a description above the parameters and you might only have to change the following parameters:

```
mail_host=localhost
mail_port=2500
mail_user=…
mail_password=…
```

Likewise for the SMS alarming („TWILIO“, „APSSMS“ and „TextAnywhere“ implementations included):

```
sms_implementation=ch.autumo.…

sms_twilio…
sms_aps…
sms_textany…
```

Note: You have to set the same „`secret_key_seed`“ as above within the configuration „`cfg/ifacex.cfg`“ too! If you are going to use the ifaceX Studio and encrypt IPC passwords, you have to define the same seed within the „General Settings“ in the ifaceX Studio.

6. Startup

As previously described, the ifaceX server executes several ifaceX modules, depending on its configuration and deployment scenario. This setup here starts the web-server for ifaceX Web administration and monitoring and the scheduler that executes Interface Processors based on their Interface Processor Configurations.

Starting the server manually or by script:

```
server.sh|.bat <start|stop|health>
```

Example:

```
server.sh|.bat start
```

On UNIX:

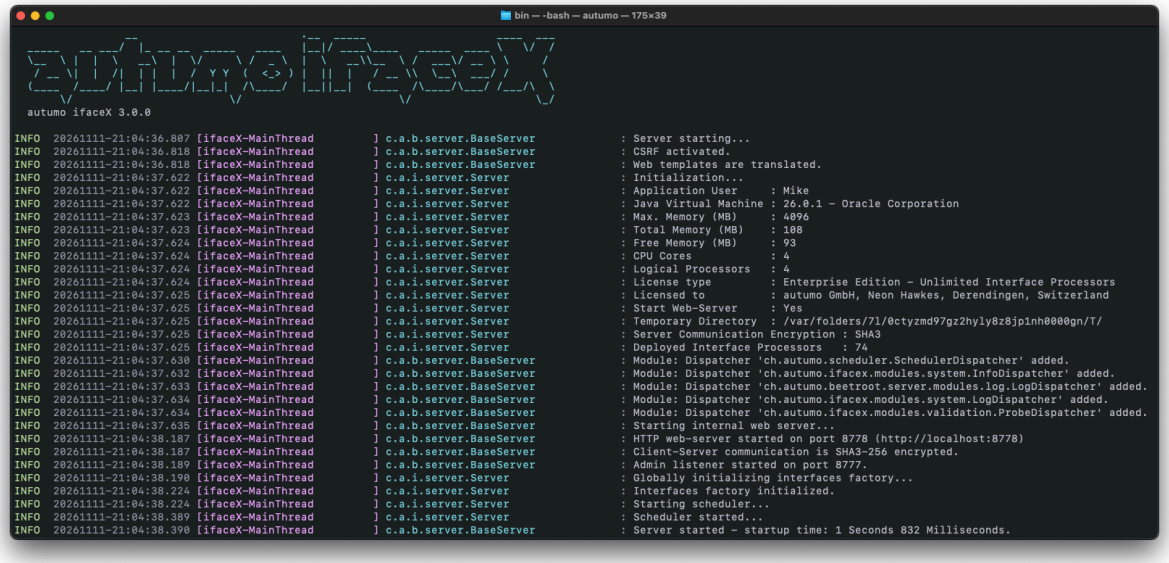


Figure 3: ifaceX Server UNIX Console Start



On Windows, the process has been stopped with „Control-Break" (CTRL-C); this case, the scheduler waits till all jobs are finished. Of course, the ifaceX Server also can be setup as a Windows Service; the necessary tools are included.

As soon the server it is started, it will immediately execute pending Interface processor Jobs that have been configured with ifaceX Web. It is essential to understand how the ifaceX Scheduler (sometimes referred as „Task Manager") respectively the Quartz Scheduler, on which ifaceX Scheduler is based, works:

If you have configured an IPC job (sometimes referred as „Task") that should be executed always in the 15th minute of every hour, and you start the server or activate it after that time, e.g., 20 minutes past 8 in the morning, it will execute based on its misfire policy (Smart, Once, Wait). Details are explained in the full ifaceX Manual.

For detailed usage of the scheduler respectively the cron-configuration (Second, Minute, Hour, Day Of Month, Month Of Year, Day Of Week, Year) of the Interface Processor Configurations, you should consult the manual. The UNIX-Cron-Syntax is compatible with the Quartz-Cron-Syntax, but the latter has more possibilities, for detailed usage see the following guide:

http://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/crontrigger.html

After the previous efforts, you'll be able to access ifaceX Web here:

http://localhost:8778 (of course „**https**" is possible too)

The username of the administrator is „`admin`" and the default password is „`ifacex`".

The server can be restarted at any time: Pending jobs (immediate queued) or jobs in execution first will be executed, before the server stops and it also takes care of possible inconsistent states by its own, e.g., when the process has been terminated forcefully! The ifaceX Server also sends alarming notifications according to the server and the respective Interface Processor Configuration.

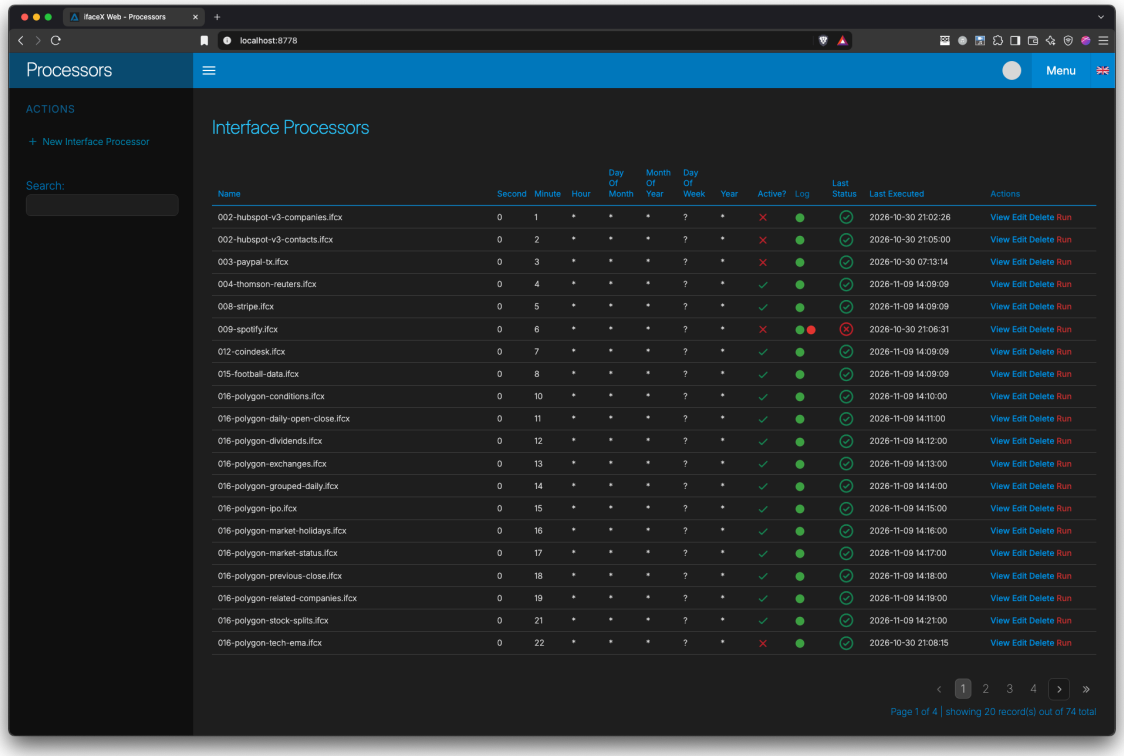


Figure 4: ifaceX Web Default View

Note: You only see „Interface Processors" in the view above, if you have configured some already.

7. Final Remarks

This minimal setup has shown you to quickly setup the ifaceX platform. Things you could do next:

- Try out the ifaceX Studio (download at https://products.autumo.ch) and load a pre-configured IPC template.
- Deploy a created IPC to the directory „`interfaces/…`“ and configure it within IfaceX Web and a cron-specification; then execute it manually or let it run by the defined point in time.
- Check out the Manual for in-depth configuration and deployment-scenarios.
- If you have purchased an Enterprise Edition License, see the „**Developer Reference**“ manual to understand how you could extend or adjust the ifaceX Platform and its operation with your own Java code.
- If you have a license: Check out the protected client area on the web for new download material and additional Interface Processor Reader/Writer packages (e.g., Clouds).

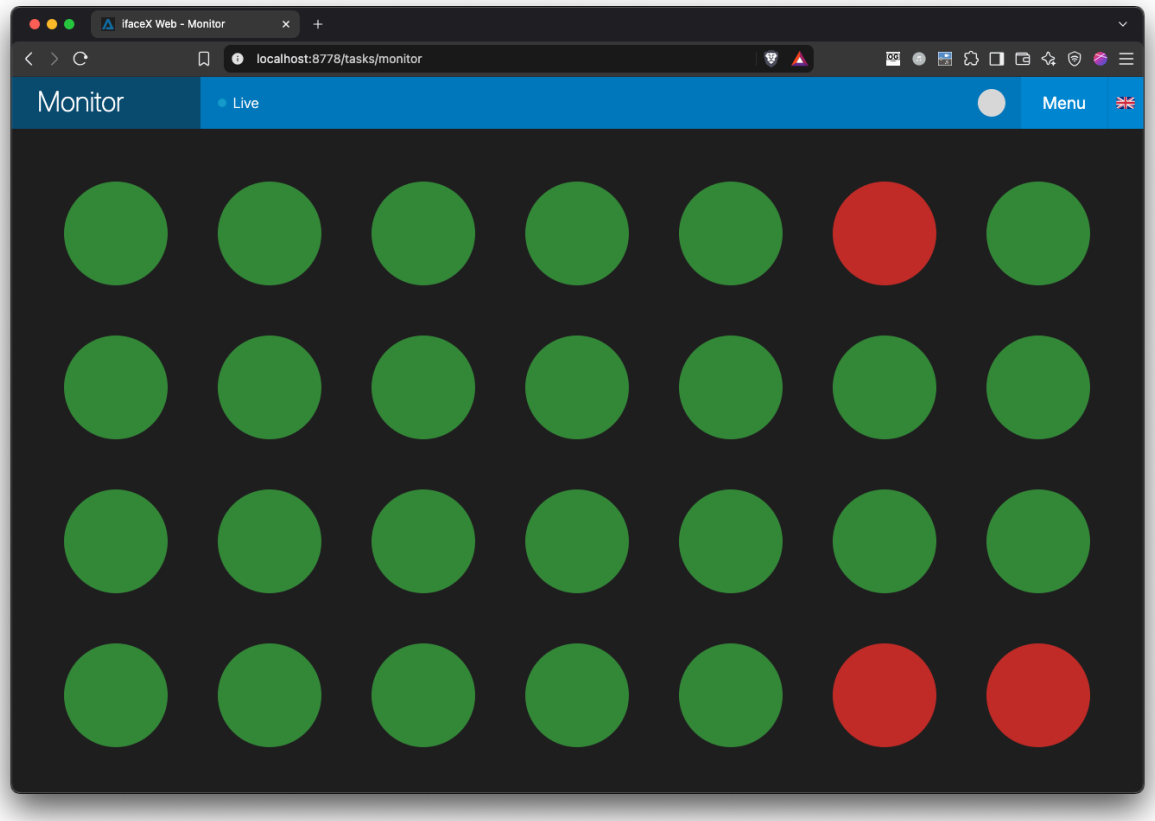


Figure 5: ifaceX Web Monitor