# Features



## Platforms and Operating Systems

JobScheduler can run on Windows and Linux operating systems. Other operating systems are supported to a limited extent, e.g. for use with JobScheduler Agents and for Agentless Scheduling.

For supported platforms see the article Which platforms is JobScheduler available for and what platform support is provided?

Read more ...

## Database Management Systems

JobScheduler uses a database to store the status of jobs, job chains and orders as well as job protocols and the job history. The database allows such information to be persistent, allowing, for example a job or order to be restored to its previous state after a JobScheduler restart or in case of failover.

Read more ...

## User Interfaces

JobScheduler comes with the following user interfaces:

- JOC Cockpit
  which provides a web-based interface for starting and monitoring jobs, job chains and orders in real-time.
- JOE - JobScheduler Object Editor
  which is used to manage JobScheduler objects such as jobs, job chains, orders, Schedules etc.
- JOC - JobScheduler Operations Center (up to release 1.10)
  which provides a web-based interface for starting and monitoring jobs, job chains and orders in real time.
- JID - JobScheduler Information Dashboard (up to release 1.10)
  which provides information about daily work plans - what will run today, what has been executed, what executions were late, what is scheduled for tomorrow etc.

Read more ...

## Command Line Operation

The JobScheduler can be operated from the command line, allowing a wide range of operations to be carried out by an external application. This includes

- checking the JobScheduler status,
- controlling the status of jobs, job chains and orders,
- adding orders to job chains,
- adding events for Event Handling.

The PowerShell CLI provides the whole range of operations, a smaller subset is available for the Unix shell.

Read more ...

## Start Times

Time events are one way of starting Jobs and Orders, with Start Times being set at a particular time of day, weekday, day of month etc. Start times can be single or repeating and can be grouped together in Schedules.

Job scheduling activities can also be limited to Time-slots:

- Jobs would have to wait for a time-slot to become available, e.g. between 08:00am and 11:00am.

Read more ...

## Logging

The JobScheduler creates a number of logs to provide specific information about jobs, job chains, orders, tasks and JobScheduler installation and operation.

Read more ...

## Jobs, Job Chains and Orders

The JobScheduler's unique job and order concept includes the organization of jobs into job chains and the use of dependencies.

- Jobs are the basic unit for the processing of executable files (programs, scripts, commands etc.).
- Job Chains can be seen as an assembly line on which a number of job nodes are passed sequentially.
- Orders represent triggers that will cause a job chain to be started according to calendar events.

Read more ....

## Scripting

The JobScheduler comes with a scripting interface that allows execution of scripts in languages such as JavaScript, etc.

- Languages that are supported by the Java javax.script package (ECMAscript compatible engines such as Rhino, Nashorn).
- Languages for Windows that are provided by PowerShell Jobs and VBScript Jobs.
- Scripts can be used to extend jobs by job scripts and monitor scripts and are a lightweight solution for individual control of processing behavior.

Read more ...

## Error Handling

The JobScheduler comes with a number of methods for error handling. These include:

- stop a job: running orders have to wait for the job to become available
- suspend an order: the order waits to be resumed later on
- setback an order: make an order repeatedly try to continue processing after a predefined time interval
- make an order leave a job chain
- make an order continue processing with a specific job node for error handling

Read more ....

## Event Handling

JobScheduler Event handling is a mechanism for implementing complex dependencies between jobs or between jobs and external events.

Read more ....

## High Availability & Fault Tolerance

JobScheduler can be operated for high availability and fault tolerance. A number of options are available:

- Resilience includes measures for operational robustness that are intended to cope with outages.
- Redundancy provides fail-over capabilities in case of outages:
    - A Passive Cluster including the set-up of a Primary JobScheduler and redundant Backup JobScheduler.
    - An Active Cluster that allows the execution of jobs on distributed server nodes.
    - The dynamic assignment of JobScheduler Agents on different server nodes.
- Recovery Strategies provide a set of measures to restore the scheduling service after an outage.

Read more ...

## Cross-Platform Scheduling

JobScheduler uses two methods for remote execution: JobScheduler Agents and Agentless Scheduling per SSH.

- Agents are used in a Master / Agent Cluster and are completely controlled by a Master JobScheduler. They are used for simplified roll-out do and not require an individual configuration and database connection.
- Remote execution by SSH does not require a JobScheduler component on the remote server - instead an existing SSH server is used to create a secure shell and to execute commands and programs.

Read more ...

## Central Configuration

Central configuration allows the efficient distribution of configuration files from a central source to distributed JobScheduler instances by use of a Supervisor JobScheduler.

The single point of configuration ensures accurate and punctual delivery of the configuration data to all instances.

Read more ...

## Directory Monitoring and File Watching

JobScheduler offers two methods to start jobs and job chains automatically based on the arrival of incoming files.

- Directory Monitoring is used to automatically start jobs when a change occurs within a specified folder. Such changes include events for addition, modification and removal of files. For each event a job is executed that can handle the files associated with this event.
- File Watching is used to automatically start jobs and job chains when files arrive in a specified folder. For each file an order is created that triggers a job chain.

Read more ...

## Managed File Transfer

JobScheduler provides two methods for managing the transfer of files:

- YADE JITL job templates which come with the JobScheduler and
- YADE Client CLI, the Command Line Interface for YADE that can be executed by shell jobs and independently from JobScheduler.

Both methods make use of the YADE implementation that comes with a number of unique features, see YADE - Features

Read more ...

## Resource Contention Management

Process classes and locks can be used to manage the use of resources such as databases or printers:

- Process Classes:
  - limit the number of jobs that are running concurrently.
  - specify remote JobScheduler Workload instances and Agents on which jobs should be executed.
- Locks:
  - limit the number of jobs that access common resources such as databases in parallel.
  - allow mutually exclusive access, i.e. making jobs wait (without consuming any CPU) for a lock to be released.

Read more ...

## Programming Interfaces

The JobScheduler comes with a number of powerful interfaces that are targeted at the following scenarios:

- implementing jobs and monitors that would e.g. check execution results and decide on specific actions. Such implementations often make use of the API Interface to check and manipulate JobScheduler Objects.
- developing individual programs and complex jobs, e.g. based on the Java API Interface, that would manipulate JobScheduler objects, e.g. create and submit orders from an individual application.

Read more ...

## Notifications

JobScheduler comes with its own mail client which it can use to send notifying e-mails in the event of, for example, jobs ending in error.

Read more ...

## Monitoring

JobScheduler can be monitored by System Monitors. Such products include e.g. HP OpenView®, Microsoft SCOM®, Nagios®, op5®, Opsview®, Zabbix® etc.

As System Monitors are restricted to check the availability and performance of a monitored service the JobScheduler provides additional functionality for System Monitors to report on individual job failure and recovery. The JobScheduler Monitoring Interface can be used with any System Monitor that provides a command line tool for passive checks.

Read more ...

## Maintenance Window Management

Maintenance Window Management ensures that for a given time slot no jobs shall be executed. JobScheduler supports a number of mechanisms to implement maintenance windows.

- Use of the `Pause` and `Continue` options with the JOC GUI or the respective XML Interface commands `<modify_spooler cmd="pause"/>` and `<modify_spooler cmd="continue"/>`.
- Script solution for Maintenance Window
- Use of Locks
- Use of Schedules

Read more ....

## Localization

Language files are provided for the installation of JobScheduler and for most of its operating interfaces. The default language is English.

Read more ...

## Reporting

The JobScheduler's reporting interface stores a configurable subset of job and order history information from any number of JobSchedulers in an open data model in a reporting database. This interface is intended to be used by users writing their own reports using pivot tables or SQL queries.

Read more ...

## Migration and Integration

JobScheduler provides a number of features to map the functionality of other job scheduling systems. JobScheduler does not include any built-in capabilities for automated migration from legacy workload automation products. However, the SOS provides Migration Services for a smooth transition.

There are integration scenarios that would allow JobScheduler to be operated with legacy scheduling products, e.g. by JobScheduler Agents.

Read more ...

## JOC Cockpit Operating and Monitoring GUI

The JobScheduler Operations Center (JOC) Cockpit is the end user interface for JobScheduler. FEATURE AVAILABILITY STARTING FROM RELEASE 1.11

Developed following the principles of “Keeping things simple” and “Minimization of clicks”, the JOC Cockpit brings a modern, responsive user interface, web service API and finely configurable authentication and authorization to Open Source job scheduling.

The JOC Cockpit completely replaces and extends the functions of the classic JOC and the JobScheduler Information Dashboard, JID, so that a single interface is sufficient for carrying out all tasks - from monitoring jobs and job chains to checking logs from the history.

Read more ...

## Editor for Creating and Managing JobScheduler Objects - JOE

JOE - the JobScheduler Object Editor is used to create and manage JobScheduler objects such as jobs, job chains, orders and schedules

Read more ...

## Library of Standard Jobs - JITL

JITL - JobScheduler Integrated Template Library - is a set of standard jobs that is shipped with JobScheduler and can be easily parameterized for your environment.

Read more ...

## Build JobScheduler from the Sources

Our software is Open Source. Therefore users are able to compile the source code of the products. This is particularly useful if you use our products on platforms with limited support, see Which platforms is JobScheduler available for and what platform support is provided?

Read more ...

**Pages**

- Reporting
    - JobScheduler Reporting Interface
- Migration and Integration
    - Controlling JobScheduler with a Tivoli Workload Scheduler Master
- GUI for Daily Work Plans - JID
    - Daily plans in JID in detail
- Operating and Monitoring GUI - JOC
    - JOC - Basic Functions List
    - JOC - Browser Support
    - JobScheduler Web-Server
    - Supported time zones
- Editor for Creating and Managing JobScheduler Objects - JOE

**Navigation**

---