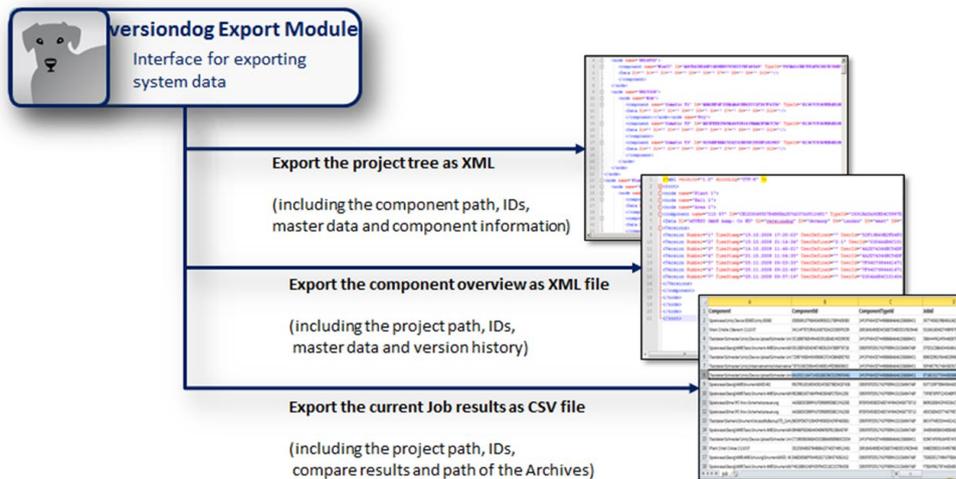


Factsheet

versiondog ExportModule



versiondog – for more Efficient Production

versiondog is the leading version control and data management software solution for automated production. It makes tracking changes and safeguarding data significantly more efficient.

versiondog brings order and clarity where project data needs to be continually changed and made available from a central source. The increased safety, security, and certainty provided by this software system quickly results in measurably increased productivity. versiondog makes it easy for you to optimize the interplay between all your different types of robots, controllers, field devices, drives, programming languages, file formats, and software applications.

This data management system gives you ultimate data traceability, minimizing your risks and costs, and saving you time and effort.

The ExportModule

The ExportModule is an add-on that is used to export data and reports from the versiondog system. With it you can export to CSV and XML whenever you need to. Data is structured for straightforward import and further processing in other systems. This open interface is one of the ways that versiondog supports the processes of analysis and optimization across multiple systems.

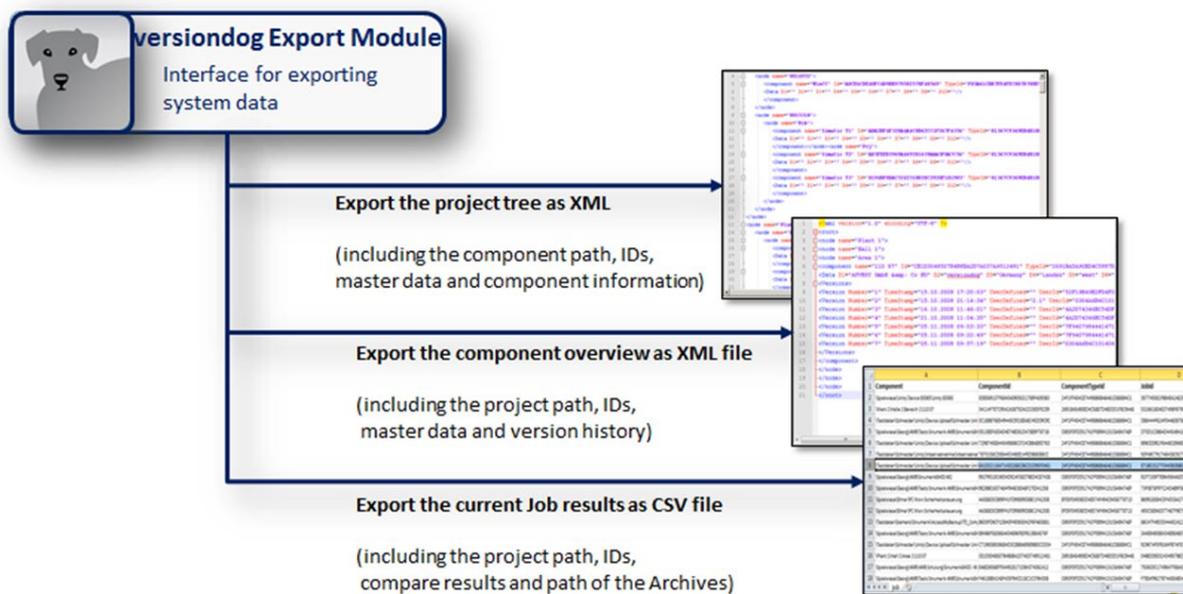


Fig. 1: The Command line controlled interface for the export of the project tree, component information, and job results.

Application Example 1

"Currentness indicator" for automated backups

The "currentness indicator" is determined from the cyclical export of job results for which a difference exists between the backup and the most recent version on the server.

From versiondog 4.0 and later exported reports can be used to determine:

- When exactly the last backup was the same as the most recent version on the server.
- How often (since the last time they were the same) a difference between the backup and the latest version was reported because a new job was executed.
- How often the comparison result could not be determined (e.g. due to changes to the firewall).

Users can now use filters to find out which PLCs have been running a different version to the server for a long period of time. Three different time allowances can be defined as follows:

- 1) Non-critical time allowance: Ongoing maintenance work or modifications always result in differences, but these can be classified as non-critical as they only last for a few days.
- 2) Time period for notifications: Differences between backups and the version on the server are displayed as notifications if they are present for longer than a certain amount of time (e.g. 1 week). In this case, it is necessary to review the situation and to Check-In the latest version onto the server as soon as possible.
- 3) Time period for critical warning: If the notification (in number 2) is not addressed then after a certain period of time (e.g. 4 weeks) it will be treated as a critical warning. In this case, the situation must be reviewed, and it must be ensured that the most recent version is both checked in on the server and loaded onto the PLC.

The "currentness indicator" for automated backups is therefore calculated as follows:

$$Q = (n - x) / n$$

where:

Q = "Currentness indicator" for automated backups

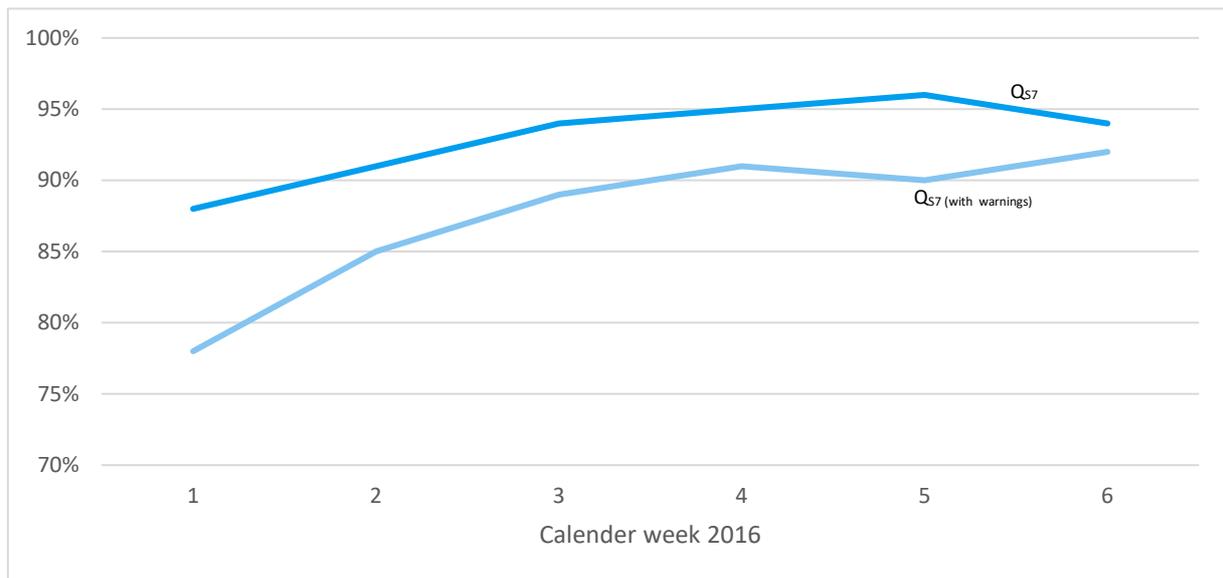
x = Number of critical warnings

n = Number of jobs executed cyclically with the compare policy "Compare backup with version".

versiondog ExportModule

The example below shows what the "currentness indicator" for automated backups for S7 controllers could look like:

Date	Number of S7 jobs in versiondog (n)	S7 jobs with errors or differences between job and server version		Quotient (Q _{S7}) (in percentage)
		Version and backup have been different for longer than 1 week.	Version and backup have been different for longer than 4 weeks (x)	
2016/W1	357	78	41	88%
2016/W2	361	51	29	91%
2016/W3	361	40	21	94%
2016/W4	362	32	18	95%
2016/W5	383	37	13	96%
2016/W6	383	27	22	94%



The graph and table above demonstrate one way how the "currentness indicator" for automated backups can be calculated.

Application Example 2

Automated email notifications for components that are under development and locked for other users for a certain period

Problem

Prior experience with versiondog has demonstrated that conflicts can be dramatically reduced if the user who has checked out a component has also been given the right to “Change lock state” in versiondog. When using this feature, it reduces the likelihood of two users making identical changes to a checked-out version of a component, while being simultaneously unaware that changes are being made to that same version of a component by the other user.

The lock state of “lock for other users”:

- Creates a ‘safety zone’ for the user who is working on a checked-out version of a component. It gives the user the time to optimize, test, and Check-In the updated version of the component back into the system. No other user may perform a Check-In of a new version of the component until the user, who has been allocated the right to lock the component for other users, is finished and has performed a Check-In.
- Functions as an advanced notification for an upcoming new version of a component. This advanced notification can be accompanied by a comment. While it is still possible to perform a Check-Out of a component, warnings and notifications make it clear that a user is currently working on the component in question and that a new version of the component will soon be available.
- Regulates and gives clarity to the workflow process and ensures that changes that have been checked in by one user cannot be overridden by another user. It prevents changes (to the last checked-out version of the component) from being checked in, until that component is unlocked.

What happens when a user who has the right to lock the component for other users appears to have stopped working on said component? The reasons for this vary. For instance, the user may have been interrupted mid-project and re-assigned a different task due to new priorities. It is possible for the administrator to take back the user’s right to lock the component for other users, but how does the administrator decide whether the user has forgotten to check the component back in or whether the user has kept the component locked for a reason?

versiondog ExportModule

Solution

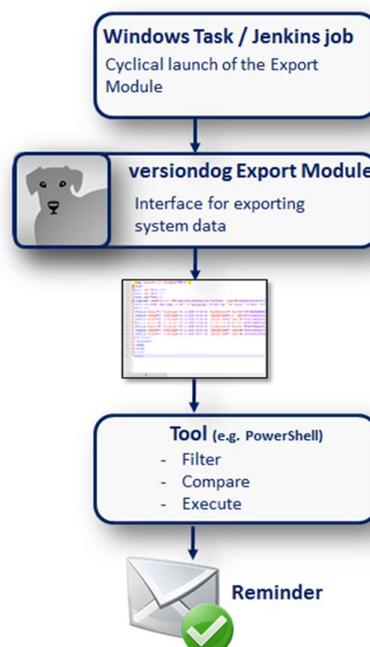
From versiondog 5.0 and later, the exported report will display the following information relevant to perform a Check-Out of a component:

- Who has the right to lock the component for other users (name, user account, email address).
- When a component was checked out and locked for other users and on which computer it was checked out from.
- A comment detailing the reasons why the component was checked out and locked for other users.

In addition to the information detailed in the report, it possible to automatically send out an email notification to all users who have been working on one or more components for an extended (configurable) period of time.

The example below demonstrates how this is done:

- A Windows task i.e. a Jenkins job (with the name “AutoReminder”) is configured to execute once a week.
- The task “AutoReminder” will create a report about the component and send it via the ExportModule.
- A tool that has been already set-up (with PowerShell, for example) will then evaluate the exported information as follows:
 - o All components with the status “locked for other users” are identified from the XML format of the exported report.
 - o It then identifies all email addresses of the users who have locked components for longer than a certain period of time (e.g. longer than two weeks).
 - o A friendly reminder is automatically sent to those email addresses.



versiondog ExportModule

Support

```
// Example of launching the ExportModule:
VDogAutoExport.exe /at:c "/rd:D:\vdClientArchive" "/CFile:D:\AutoReminder\export.ini"

// with the export.ini:
[Common]
ReportType=ComponentTree
ExportFile=D\AutoReminder\ExclusiveEdit.xml
WithVersions=N
WithMasterData=N
EnableNodeTree=N
[User]
Account=versiondog
Password=change-it
```

```
// Example of the task (ExclusiveEdit.xml):
<?xml version="1.0" encoding="UTF-8" ?>
<root Timestamp="...">
<component name="..." Path="..." Id="..." TypeId="..." State="...">
</component>
<component ...
</component>
...
<component name="..." Path="..." Id="..." TypeId="..." State="...">
<LockState
  State="1"
  UserId="6D0DD6948E4B4E87A763C12744B0E7BA"
  UserName="MyDomain\MaxMustermann [Max Mustermann]"
  EmailAddress="max.mustermann@myDomain.de"
  TimeStamp="14.03.2017 11:15:32"
  Days="20"
  ComputerName="VMWARE.WIN7-2-GK"
  Comment="Test">
</LockState>
</component>
...
</root>
```

XML-Tag	Description
<LockState>	The <i>LockState</i> -Tag contains information relative to the lock state of a component. It is only available when the lock state is activated.
<LockState State="n">	The attribute <i>State</i> describes the lock state: 1: Lock for other users 2: Under development
<LockState EmailAddress="...">	The <i>EmailAddress</i> attribute contains the email addresses of the users who have locked the components. If no address has been defined, an empty character sequence will be registered.
<LockState TimeStamp="...">	The <i>TimeStamp</i> attribute describes time (UTC) when the component was locked (as a character sequence in the format of the operating system). Example: „09.10.2015 08:32:56“
<LockState Days="n">	The <i>Days</i> attribute describes the number of days that have gone by since the component was locked.

More information can be found in our versiondog ExportModule [Online-Tutorial](#). If you have any questions or comments, please do not hesitate to contact our support team at Support@auvesy.com.

versiondog ExportModule

Features and functions

Interface via Command Line	✓
Export of project structure (inc. path), component type and master data	✓
Export of component information (inc. path), component type, master data and version history.	✓
Export of a list of all jobs together with the associated components, component types and latest job results.	✓
Export as an XML or CSV file for straightforward import into other systems	✓
Access rights taken into account when exporting	✓
Ensured compatibility with future versions	✓
Ongoing development and enhancement	✓
Tutorial available in versiondog INFO	✓

System Requirements

For current system requirements, please visit versiondog INFO at www.versiondog.info/en or contact us at info@auvesy.com.

More Information

AUVESY Inc

Tel. +1.616.888.3770

Email info@auvesy.com

Web www.auvesy.us