independIT Integrative Technologies GmbH

New Features Release 2.11

March 21, 2024

Notification

Zope 2 frontend Since the Zope 2 frontend is based on Python 2 and this version of Python has not been supported for a long time, we will discontinue support for Zope 2 from the next release (2.12).
The current release 2.11 also contains an SDMS.zexp and some Python scripts. This means that an existing Zope 2 installation can continue working with the 2.11 server. However, it is no longer possible to reinstall Zope 2.

New features

New frontend A new frontend based on NodeJS was developed from scratch. The browser-based application has been designed as a single page application. This eliminated a significant downside of the Zope-based frontend. Using it is now also considerably more fluid and pleasant due to the lack of round-trip dialog steps. The application has a much more modern look as well.
In this release, the new GUI is still being supplied as alpha software. Although intensive testing has uncovered many of the teething problems, it cannot be ruled out that some functions may not be 100% perfect. The Zope-based frontend should therefore be kept at least as a fallback solution.

Triggered By Row Limit Some jobs are triggered by many other jobs. This applies, for example, to jobs that send emails or other types of notifications in the event of an error.
It is now possible to specify a limit in the LIST TRIGGER statement so that only some of the triggers are displayed. This significantly improves the performance.
A limit is generally active in the frontend, TB_ROW_LIMIT, with 100 as the default setting.

ps Utility for Windows The standard Windows program WMIC was used to determine the list of all the processes with their start times. However, it transpired that problems arise with the switch from summer time to winter time or vice versa. An alternative to WMIC was created with the aid of PowerShell. This was very slow, however, and led to excessive CPU consumption.
A third solution, "winps", was therefore written in C based on the *ps* utility under Unix. This solution has no problems with the switch from or to DST and is actually even faster than WMIC.

Better frontend support for object monitoring Watch Types and Object Monitors can be defined from the GUI. The manage_watchtype privilege is required for this.

Build hash and build date As support requests often ask for the build hash and this had to be determined in a somewhat cumbersome manner, the build hash is now outputted with the show system. And since the hash value is not particularly meaningful for humans, the build date is outputted as well.

Java compatibility Newer Java versions (11, 13, 17 and higher) are now also supported. However, the MEMFLAGS parameter in the java.conf file must be changed when using Java 17 and higher.
The setting

```
MEMFLAGS="-XX:+UseZGC -XX:-ZUncommit"
```

has proven itself so far.
As always, we are grateful for suggestions and comments.

Visibility of intervals Independent intervals, i.e. intervals that do not belong to any Job Definition, are now visible and usable for everyone. The other privileges, DROP and EDIT, are naturally not released for general use.

Unresolved parameters It is now possible to configure how the server should handle problematic parameters. If a parameter is not found, either an error is outputted, an empty string is returned as the value, or the parameter is not replaced at all.
The configuration parameter is called *UnresolvedParameterHandling* and has the following possible values:

- ERROR – Default, a non-resolvable parameter is considered an error
- BLANK – A non-resolvable parameter is replaced with an empty string similar to in a shell
- ECHO – The parameter is not replaced

jsctl utility Instead of JSCTL.BAT the new executable jsctl.exe is used on Windows systems to install or modify the jobserver service. This was necessary because of the different treatment and necessity of quoting of Windows releases and their language settings.

Show Interval With the Show Interval command, the generated blocks can be optionally outputted for a period. This facilitates the development of complex intervals.

Expand Rule An Expand Rule for the Enable Interval from the parent-child relationship of two Job Definitions was added to the dump.

Configurable ticket interval The ticket interval can now be configured. This can speed up the takeover of operation by a warm standby server as it waits for a ticket interval three times. The minimum time between two ticker updates is three seconds. The default value is now 20 seconds.

Broken Finished Handling A server parameter BrokenFinishedHandling with the possible values RESTRICTED and CLASSICAL was added to define how the server behaves if a job gets a BROKEN_FINISHED status.
CLASSICAL is the original behaviour that BROKEN_FINISHED considered to be a Broken State. In this case, the Exit Status of the job is set to Broken State, which in turn can result in further processing.
With RESTRICTED, the Exit Status is not (automatically) set. This gives you more control over what happens.

List Calendar LIST CALENDAR can now also filter by owner

scrolllog options The semantics of the *-f* option has changed. Log output is not only written to the log files but also to stdout.
This option is used by *sdmsctl run*. This allows the server to be started in the foreground for debugging purposes.

Show Folder and Show Scope parameter display The Show Folder and Show Scope command now shows all visible parameters, including in particular the parameters that are defined at parent level (or higher).

Parameter EXITCODE EXITCODE was added as a special parameter for jobs. This can be used, for example, to create more precise error messages in notifications.

New parameter type The parameter type IMPORT_UNRESOLVED was added. These parameters are evaluated in the context of the importing job and not, as with IMPORT, in the context of the defining job.

List scheduled command A new command *LIST SCHEDULED* has been implemented. This can be used to determine when which batches and jobs are to be run for the specified period. As in the calendar, you can also filter by job name.

Disabled job servers A job server can be disabled at any time. However, it will still be able to connect to the server until there are no more active jobs left. Access is then blocked.

Jobserver connected display Online job servers are immediately displayed in red if they do not have a server connection. This eliminates the occasional misleading display where the GUI continues to show the job server for a while in the expectation that it will reconnect immediately anyway.

Compact list with Resource State Mappings The statement LIST RESOURCE STATE MAPPING FOR *profileName*; only shows the mappings that are compatible with the specified Resource State Profile.

Listen interface Until now, all interfaces have always been listened to. A specific interface can be given with the parameter *Interface* or *SSLInterface*.

sdmsh comfort feature In *sdmsh*, the last statement in a multicommand does not have to be closed with a semicolon. sdmsh recognises this and automatically inserts the missing character. However, according to the language's official syntax the semicolon is still necessary. The feature is therefore used exclusively for convenient handling of sdmsh.

Extension of the Connect command The supplement ZERO TERMINATED can be specified for the JSON, PYTHON and PERL protocols. The output structure that is returned in response to a command is then terminated with a null byte. This extension makes it much easier to read in the response.

Condensed output The listing of the Job Definition hierarchy and dependency hierarchy can be condensed. Columns that are generally not required are removed from the output. This results in better performance because less data has to be processed.

Server IP address In warm standby operation (i.e. when a second server is started which immediately takes over when the active server fails), it is not always obvious which server is currently active.
The IP address of the active server is now saved in the REPOSITORY_LOCK table. This information can also help to rapidly troubleshoot errors that have occurred accidentally (such as an error in the database configuration). Job server configuration

New job server parameters There are three further parameters for the job server configuration.
In some environments, the synchronisation with the start times for the processes does not function perfectly. Occasionally, the determined start time deviates too much from the actual start time. As a result, the job is moved to the status BROKEN_FINISHED even though it is still running. This can be prevented by increasing the tolerance when comparing the two start times. The **STARTIME_JITTER**

parameter (default: 5 seconds) can be set to a higher value for this purpose. If it is set to 0, the start time is no longer taken into account. Any process with the same PID as the original process is interpreted as being the process that is to be monitored.
This is usually harmless. To cause a misinterpretation, the PID must be reassigned by the operating system extremely quickly.
Due to the varying ways in which different operating systems interpret the correct termination of a line (Unix only requires a line feed, Windows requires a carriage return as well), the wrong termination could be used in the RUN_PROGRAM. This then leads to errors in how the arguments are interpreted.
As the job server runs on either a Unix-like or a Windows system (but never both), the job server can carry out a conversion to either line feed or carriage return line feed.
For this, the parameter **CONVERT_NEWLINE** is set to 0, the default if unset, and a replacement does not take place. The value 1 results in a conversion to line feed. The value 2 is used for converting to carriage return line feed.
The third parameter, **HTTP_INTERFACE**, is used to determine which interface should be accessed by the job server to display the log files. By default, all the interfaces are listened to. This can be reduced to one specific interface using the stated parameter.

Rename parameters If parameters are set in the Alter Job command, they can be renamed by stating the parameter ID. The references are preserved (REFERENCE and CHILDREFERENCE parameters).

Cancel with Kill The Kill option can be specified to cancel jobs. All running jobs with a Kill program are then first killed and subsequently cancelled.
A Kill recursive was required for the implementation. This operation can also be used without Cancel.

UTF-8 encoding for Log Output When Log Output is returned, the text is converted to UTF-8. This allows special characters to be displayed correctly.

Fixed errors

Quotation marks for parameters and trigger names Missing quotation marks for triggers and parameter names caused problems with lowercase names and special characters. These names are now quoted correctly.

Upgrade scripts improved The generated_upgrade scripts now no longer contain drop statements for views that will only be used in later releases. No attempt is made to create these views if the base tables do not yet exist.

Zope 5 support Some incompatibilities with newer Zope 5 releases have been removed.

Resource Schedule as INTERNAL If audit entries are made during the Resource Scheduling triggered by a Get Next Job request from a job server, this is executed as User INTERNAL.

Parameter resolution In some (incorrect) constructions of mutually referring parameters, unclear error messages could occur. These have been improved.

Copy Job Definition The comments were not copied when cloning Job Definitions. This error has been fixed.

Missing status in the Exit State Profile If the Timeout Status is not included in the Exit State Profile for a job, the job is set to ERROR and no more exceptions are thrown.
If the Limit State is not included in the Exit State Profile for a job, the job is set to ERROR. The Broken State of the Exit State Profile is not set as this can cause the trigger that has reached its limit to be reactivated.

Correct handling of incorrect syntax Untimed parameter references at the end of a string, such as '${PARM', result in an ArrayOutOfBounds exception.

Empty SELECT lists An empty SELECT list is valid in some database systems, but could cause the server to crash. This is now intercepted properly.

Buffer for PATH variable Only 1 KB was provided for the content of the PATH variable. This has been increased to 8 KB.

Commit at the end of an upgrade A Commit statement is now written at the end of the generated_upgrade scripts to prevent some database engines from rolling back the entire transaction (i.e. the script) because the Commit is missing.

Race condition If a BROKEN_FINISHED situation is detected, the system waits one second to rule out the possibility that the situation was detected incorrectly.

Edit Resource Requests An existing Resource State Mapping can also be removed again in a Resource Requirement.

Quote identifiers All identifiers are now quoted in the SQL statements to avoid problems with new SQL keywords.

Create User The system waits 1 ms when a user is created. This ensures that the generated salt is different for each user, even when it is created automatically.

Zombies under Windows Prevent the creation of zombies under Windows; references to processes are explicitly deleted to prevent zombies.

Hide deleted objects Deleted groups and deleted users are no longer displayed.