

1 Cronology 4.2 New Features Overview

1.1 Key New Features

For existing 4.1 users who are already familiar with Cronology, release 4.2 brings many new features, the following however are worthy of immediate note:

- **Remote Jobs:** in today's IT environments it is increasingly unlikely that application code can reside on the same server the database is running on. Cronology now allows jobs to be run remotely on different servers over SSH.
- **File Watchers:** A new job type of File Watcher has been introduced, allowing jobs to be triggered by the arrival / delivery of an OS file
- **All New Console:** the Cronology Console is now web based and has been rewritten using Oracle APEX for a cleaner more modern look and feel
- **Oracle PDB Support:** Cronology is fully supported to run in a PDB of an Oracle Multitenant database
- **Oracle Versions:** Release 4.2 is supported on all versions of Oracle 11.2.0.4 and above. Release 4.1 is only supported up to Oracle 12.1

1.2 Other New Features

- **Additional Job Source Types**
As well as file watchers, the following job source types are now also provided:
 - Python
 - Perl
- **Additional Parameter Data Types**
To better support Oracle stored procedure jobs the following parameter data types are now provided:
 - Timestamp
 - Boolean
- **Choice of E-mail Formats**
E-mails generated by Cronology can now be either:
 - HTML - formatted with a mono spaced font
 - Plain text - padded such that Outlook handles line breaks correctly
 - Plain text - pre 4.2 format
- **Console Environment Indicators**
Visual cues can be important when switching between environments. The console header can optionally be configured to:
 - Display an environment label e.g. Development, QA, Production
 - Be colour coded e.g. red for production, blue for development

- **More Flexible Frequency Definitions**

The pre-defined job frequencies at the MIN, HOUR, DAY, WEEK, MONTH and YEAR level have been removed and can now be defined to any value ranging from 1-999. In previous releases the pre-defined set of frequencies was limited e.g. 5 MINS, 10 MINS, 30 MINS etc. If a bespoke frequency e.g. every 3 weeks was required this had to be defined using one of the other frequency types (SQL, CALENDAR or CRON). The new method allows much greater flexibility defining job frequencies.

- **Delayed Job Poller Shutdown**

The facility now exists to shut the job poller down at a pre-determined time in the future – the job poller status will appear light green if a pending shutdown has been requested.

- **Maximum Success Exit Code**

For jobs that reply on a UNIX exit code (e.g. shell scripts – or in certain cases SQL scripts that use WHENEVER SQLERROR EXIT n) you can now specify the maximum exit code that is deemed successful completion. In previous releases success was an exit code of 0 and anything else (1 or above) was a failure – there is now the flexibility to specify the maximum code that indicates successful completion. The exit code for every script (success or failure) is now stored in the Information field of the job run history.

- **Overdue Behaviour**

If a job becomes overdue for execution you can now specify how you want to Cronology to handle the job once it becomes valid for execution once again:

- Catch Up – the default pre-4.2 behaviour, Cronology will execute the job the same number of times successively as would have run according to the schedule e.g. if the job executes every 5 minutes, and it has been overdue for 30 minutes it will execute 6 times consecutively
- Advance – Cronology will evaluate the next scheduled run time due after the current time and will NOT execute catch up runs

- **Easily Create Alert / Monitoring Type Jobs**

A new option to only send a completion e-mail if the job actually creates output allows for alert / monitoring type jobs to be created easily. If the job completes and creates some output in the job log (e.g. testing or looking for a certain conditions in the application or database) then an e-mail of the log content will be sent. If the job completes but produces no output then no e-mail is sent.

- **New Job Override Option to COMPLETE But Halt Execution**

A new option on job overrides allows for jobs to be marked as COMPLETE but will not launch any child jobs. Useful if you do not want to raise a job failure and also do not want subsequent child jobs to run.

- **New ENVVAR Parameter Type**

Previously UNIX environment variables could be used but had to be stored in a CHARACTER parameter. If a character parameter genuinely needed to contain a \$ character this led to issues. By creating a new parameter type environment variables are now clearly segregated in the job configuration.

- **AES-256 Password Encryption**

Improved security, any passwords stored with job definitions are now AES-256 encrypted.

cronology

Version 4.2 - New Features

- **Improved Usability**

- Job creation and amend screen redesigned for ease of use, clearer layout using a tabbed style layout for Parameters, Overrides, Messaging options etc.
- Job Status Overrides has a new Match / Does not Match condition for greater flexibility when specifying overrides
- More meaningful error messages when saving jobs and parameters – in certain areas error messages lacked clarity – this have been improved when validating changes to jobs and parameters
- Main console screen now has ability to filter by Frequency
- When bringing jobs online the display lists the jobs in hierarchical format for ease of viewing
- All generated export scripts and documentation now include a section detailing which filters were in place at the time the script was generated
- When testing SQL parameters, if the parameter contains an Oracle sequence the user is asked if they actually want to execute the statement or perform a syntax check only (to avoid advancing the sequence)

- **Improved Stability**

- A new configurable system parameter PARAM_EVAL_LIMIT_SECS now exists which stipulates the maximum amount of time allowed when evaluating an individual job parameter. Because Cronology allows SQL based job parameters i.e. parameters that are the result of SQL statement it was possible to have parameters that could take a long time to evaluate. This parameter helps detect and prevent this, parameters that exceed this new limit (default is 2 seconds) will cause the corresponding jobs to fail execution
- New configurable HOST_NAME system parameter. This stores the name of the host the Oracle instance running Cronology. This safety feature prevents Cronology running on an environment before it is reviewed. E.g. if the server is upgraded or the installation is copied to another environment and the host name changes then this will cause a conflict with this parameter. Cronology will not launch until the HOST_NAME parameter is reset. This can be invaluable if you have cloned your environment (e.g. from production) to another server but the database connect strings specified for remote jobs are no longer applicable for that environment
- Additional process checks for background processes - was already checking for dead processes, now checking both ways i.e. status is STOPPED but the process is actually running. Status' are adjusted accordingly

- **Improved Performance**

- Optional asynchronous commit processing for Log Writer – useful if jobs are producing large amounts of log output (new system parameter LGWR_ASYNC_COMMIT)
- Numerous internal enhancements to improve job polling and job launching efficiency

- **Improved Maintenance**

The main JOB_LOG database table may now be partitioned by date (in house Oracle license permitting) to aid the maintenance of this table. Options to delete in batches and shrink storage are also provided.

- **Improved Code Tracking**

All Cronology server and console code is now tracked to three or four digits. The System Information shown on the About screen of the Cronology Console details the code versions. This greatly aids support queries.

2 Cronology 4.2 Bug Fixes

2.1 Server

- API DELETE_HISTORY now checks if the last status change date time for a process is older than the delete date - if it is then the delete for BG_PROCESS_HISTORY is skipped
- if NOMATCH specified on JOB_STATUS_OVERRIDES still marked job as failed
- all TCP connections closed after e-mails / socket clients calls - seen max open port errors if email fails repeatedly
- smtp_mail - sender is now no-reply@cronology.co.uk i.e. valid email address syntax - for compatibility with later versions of Microsoft exchange server
- if job wait or launch error occurred the start datetime and duration could be left null, now start time is set to the submit time if no start time is captured
- parameters that contain single or double quotes are now handled correctly
- frequencies MON-FRI, SUN-THUR, SAT-SUN failed to calculate correctly
- SQL/SPROC jobs and background process password could not contain the \$ symbol
- stored procedure jobs would fail to run if no \$HOME env variable set up for 'oracle' user on DB server
- Stop potential infinite loop / recursion in find_process (internal) added NVL
- FIND_JOB_SUB_DTTM possible loss of logging if process not registered
- potential loop in FIND_JOB_SUB_DTTM fixed
- incorrect data type validation when adding parameter dependencies
- if shared server (and a dispatcher) processes running would see ORA-29516: Aurora assertion failure: Assertion failure at joevm.c:5708 process trying to survive end of call in MTS mode when executing jobs from the console – all OS calls fixed