# cronology

Version 4.2

## Installation, Upgrade and Administration Guide

Document Version 4.2.7.19


www.cronology.co.uk

# 1 System Requirements

The Cronology software is comprised of two components:

- Cronology Server – database tables, packages, sequences etc.
- Cronology Console – an Oracle APEX application to provide GUI access for users

Both components need to be installed in the target Oracle DB.

## 1.1 Cronology Server

**Supported Unix Platforms:**

Cronology Server is supported on Oracle DBs running on:

- Solaris
- Tru 64 / OSF1
- HP-UX
- Linux (OEL)

If you wish to use SNMP messaging you must ensure the relevant SNMP software is installed on your UNIX platform.

**RDBMS Requirements:**

- Oracle RDBMS 11.2.0.4 or above with Oracle APEX 19.2 installed – Oracle **APEX must be installed and configured BEFORE the Cronology server installation / upgrade begins**
- Server installations may be done remotely over SQL*Net:
    - ensure all connect strings are working properly (i.e. configured in tnsnames.ora)
    - connections as SYS are required for the install, please ensure an oracle password file exists and the database remote_login_passwordfile parameter is not set to NONE
- The target Oracle database must have the Java option installed
- Oracles initialisation parameters
    - **java_soft_sessionspace_limit**=0 (meaning default)
    - **java_max_sessionspace_size**=0 (meaning default)

**IMPORTANT:**
If server application code (e.g. SQL scripts / shell scripts) reside on the DB server, they **MUST** be readable by the Unix account the Oracle database is running from (typically 'oracle')

**Supported Cronology Server Versions Valid For Upgrade:**

- 4.1 only

**The 4.2 upgrade can only be applied to an existing 4.1 installation.** If you are running a version of Cronology below 4.1 you must first upgrade to 4.1 before upgrading to 4.2.

### 1.1.1 Multitenant Databases

Cronology should be installed one (or many) PDBs of a multitenant database. **Do not install Cronology into the CDB of a multitenant database.**

## 1.2 Cronology Console

The Cronology console requires an installation of Oracle APEX 19.2 in order to run. As per Oracle recommendations, it is advisable that a corresponding Oracle Rest Data Services (ORDs) install is used for serving APEX applications, however, if you choose to you may use the Embedded PL/SQL Gateway (EPG) to serve Cronology as typically the number of concurrent users on any given environment is relatively low.

**Please install APEX 19.2 together with the latest patch set bundle and ensure access is working from a browser before starting the Cronology installation. See Appendix F for basic install and patching instructions.**

**APEX 19.2:**
https://www.oracle.com/tools/downloads/apex-192-downloads.html

**Latest APEX 19.2 patch set bundle:**
https://support.oracle.com/epmos/faces/PatchDetail?patchId=30392181

**ORDS:**
https://www.oracle.com/database/technologies/appdev/rest-data-services-downloads.html

The Cronology Console uses Oracle's CDN (Content Delivery Network) for its static APEX resources**. Therefore configuring ORDS with the APEX 19.2 static resource files is not strictly necessary.**

More information on CDNs:
https://blogs.oracle.com/apex/announcing-oracle-apex-static-resources-on-oracle-content-delivery-network

### 1.2.1 Multitenant Databases

**If you are running your Oracle database with the multitenant architecture, rather than installing APEX into the CDB, we strongly recommend you install APEX into each PDB that requires Cronology.**

# 2 Cronology Server Installation

## 2.1 Prerequisites

### 2.1.1 Installation Variables

During the installation you will need to specify the following information:

**Company Name (max 32 chars - case insensitive)**
Your company name - this information is stored as part of the license key information.

**Database TNS Alias (case insensitive)**
Leave null ('') for a local install (assuming your Oracle environment, ORACLE_SID etc are set correctly) - or specify a TNS alias for a remote installation.

**SYS Password (case sensitive if SEC_CASE_SENSITIVE_LOGON = TRUE)**
The password for the SYS account - required for granting privileges on objects/packages owned by SYS.

**CRONOLOGY Password (case sensitive if SEC_CASE_SENSITIVE_LOGON = TRUE)**
The required password for the CRONOLOGY schema. This schema owns all the CRONOLOGY tables and packages. It is a highly privileged account - its password should be set accordingly.

**Hide UNIX Commands (ON | OFF - case sensitive)**
Determines whether CRONOLOGY operating system commands are visible on a ps listing. A value of ON means commands are not visible, OFF means commands are visible. The recommended setting is ON. WARNING! A setting of OFF may result in database logon details for running jobs becoming visible via a UNIX ps process listing.

**Java Executable Directory (case sensitive)**
The full path and filename of the java executable used for running Java applications. This is only used if Java is not configured in the application ENVIRONMENT_SCRIPT. If you are unsure enter the following command at the UNIX command prompt: unalias -a ; dirname `which java`

**Temp Directory (case sensitive)**
The directory location of the temp directory on the server (usually /tmp)

**UNIX Command Directory (case sensitive)**
The directory location of the UNIX commands on the server. If you are unsure enter the following command at the UNIX command prompt: unalias -a ; dirname `which ls`

**Partition JOB_LOGS (YES | NO - case sensitive)**
If you are licensed to use Oracle partitioning, optionally partition the JOB_LOGS table to improve housekeeping of the logging table. Values are YES or NO

Please have this information to hand **before** you begin the installation.

## 2.2 Installation

The installation should be undertaken by a DBA.

### 2.2.1 Copy Installation Media

- logon to the Unix server
- create an installation directory and cd into it
- copy the cronology_4200.zip file into the directory (N.B. the zip file should be FTP'd in **BINARY** mode)
- unzip cronology_4200.zip
- cd into the 'server' directory

### 2.2.2 Create and Edit Installation Variables Script

Rather than prompt the user for input during the installation, a script containing all required installation parameters must be setup before running the installation. This is useful as different parameter set ups can be saved for future use / reference.

- copy the install_vars_MASTER.sql script to a name of your choosing. You may choose whatever naming convention you like for the file but typically the name would include the server and database name the install is relevant for
- edit the file and enter all the relevant details – the template gives descriptions and examples for each parameter – save the file

### 2.2.3 Run Installation

To run the installation:

- logon to the target database / PDB via SQL*Plus (any user will do as the installation will reconnect as required)
- run the installation script providing your variable script name as a parameter:

  @install <your variable script name>

- be sure to read the 'Important Information' displayed when the installation completes

The installation should complete without any errors. If the install fails for any reason you should:

- look to address any database related issues (contact Cronology support if needs be)
- uninstall (see section 5.1)
- re-run the install

### 2.2.4 Set-Up Optional Features

Use the CRONOLOGY.ADMIN package (see Appendix A) to set up any additional features you require:

- SET_SMTP_SETTINGS: Set your SMTP server details if you wish Cronology to use e-mail messaging
- SET_SOCKET_SETTINGS: Set your socket server details if you wish Cronology to use socket messaging
- SET_SNMP_EXE_DIR: Identify the location of the snmptrap (snmp_trapsnd for Tru64) executable if you wish Cronology to use SNMP trap messaging
- SET_MESSAGE_TEMPLATE: If you wish to define your own message templates for bespoke, socket or SNMP messaging (mandatory for SNMP messaging)
- LOG_WRITER_ASYNC_COMMIT: Make the Log Writer use asynchronous commits when writing job log information to the database. Default is OFF
- PARAM_EVAL_LIMIT_SECS: Maximum time in seconds for parameter evaluation. Default is 5 seconds
- SET_SSH_EXE_DIR: Identify the location of the ssh executable if you wish to set up jobs that execute scripts on remote servers (see Appendix E for more details on SSH configuration)

### 2.2.5 Console User Grants

This can be done logged on as a DBA or the CRONOLOGY schema. Grant CRONOLOGY_READONLY, CRONOLOGY_OPERATOR or CRONOLOGY_ADMINISRATOR role to required database user accounts to provide Cronology Console access. N.B. Users granted CRONOLOGY_ADMINISTRATOR may grant other database users console access via the Cronology Console.

### 2.2.6 Obtain License Key

If you do not already have a valid license key for this database obtain one from Cronology Support by emailing support@cronology.co.uk – specify the database name / database ID displayed under 'Important Information' when the installation has completed.

Once obtained, the key can be set by either:

- a CRONOLOGY_ADMINISTRATOR user via the Cronology Console (using the About screen)
- a DBA using the procedure specified in Appendix D

# 3 Cronology Server Upgrade

**The 4.2 upgrade can only be applied to an existing 4.1 installation.** If you are running a version of Cronology below 4.1 you must first upgrade to 4.1 before upgrading to 4.2.

If you are upgrading Cronology as part of an Oracle database upgrade it is highly recommended you upgrade Cronology to 4.2 **BEFORE** the Oracle database itself is upgraded.

If you choose to upgrade Cronology **AFTER** an Oracle database upgrade **AND** you have changed the database name / id your license will be invalid. Please obtain a new licence key from Cronology Support (support@cronology.co.uk) **BEFORE** you start the Cronology upgrade.

## 3.1 Prerequisites

### 3.1.1 Upgrade Variables

During the upgrade you will need to specify the following information:

**Database TNS Alias (case insensitive)**
Leave null ('') for a local upgrade (assuming your Oracle environment, ORACLE_SID etc are set correctly) - or specify a TNS alias for a remote upgrade.

**SYS Password (case sensitive if SEC_CASE_SENSITIVE_LOGON = TRUE)**
The password for the SYS account - required for granting privileges on objects/packages owned by SYS.

**CRONOLOGY Password (case sensitive if SEC_CASE_SENSITIVE_LOGON = TRUE)**
The password for the existing CRONOLOGY schema.

**Partition JOB_LOGS**
The Cronology JOB_LOGS table can be partitioned (if you have the required Oracle licenses) for ease of maintenance. The upgrade can perform this partitioning for you by setting this parameter. Valid values are 'YES' or 'NO'

**Check Current Install State**
Leave this parameter as 'YES' unless you experiencing upgrade issues. Valid values are 'YES' or 'NO'. There may be scenarios where you have upgraded your Oracle database before upgrading Cronology and although the Cronology installation is still valid, it is unable to run correctly, this will interfere with the upgrade process.

If you have upgraded your database to Oracle 18c or above before upgrading Cronology, i.e. the current Cronology installation is still 4.1 and your database is 18c or above, then set this parameter to NO.

**New License Key**
If you are upgrading Cronology on a new database (i.e. the 4.1 installation was imported from another database) you may need a new license key. Please email support@cronology.co.uk with your existing licence key entries to determine if a new licence key is required:

```
select name, value
from   cronology.parameters
where  name like 'LICENSE_KEY_PART%'
and    param_id < 0;
```

## 3.2 Upgrade

The upgrade should be undertaken by a DBA.

### 3.2.1 Copy Installation Media

- logon to the Unix server
- create an installation directory and cd into it
- copy the cronology_4200.zip file into the directory (N.B. the zip file should be FTP'd in **BINARY** mode)
- unzip cronology_4200.zip
- cd into the 'server' directory

### 3.2.2 Create and Edit Upgrade Variables Script

Rather than prompt the user for input during the upgrade, a script containing all required upgrade parameters must be setup before running the upgrade. This is useful as different parameter set ups can be saved for reference.

- copy the upgrade_vars_MASTER.sql script to a name of your choosing. You may choose whatever naming convention you like for the file but typically the name would include the server and database name the upgrade is relevant for
- edit the file and enter all the relevant details – the template gives descriptions and examples for each parameter
- validate your entries and save the file

### 3.2.3 Run Upgrade

As part of the upgrade you will be asked if you wish to backup your existing Cronology installation – **it is highly recommended you do this** in case you need to restore / rollback the upgrade.

To run the upgrade:

- logon to the database / PDB via SQL*Plus (any user will do as the installation will reconnect as required)
- run the installation script providing your variable script name as a parameter:

  @upgrade <your variable script name>

- be sure to read the 'Important Information' displayed when the installation completes

The upgrade should complete without any errors. If the upgrade fails for any reason you should:

- look to address any database related issues (contact Cronology support if needs be)
- apply any new license details if you have changed the database name / id
- re-run the upgrade script (it may be run as many times as necessary until it completes successfully)

### 3.2.4 Set-Up Optional Features

Use the CRONOLOGY.ADMIN package (see Appendix A) to set up any additional features you require:

- LOG_WRITER_ASYNC_COMMIT: For Oracle 10.2 and above make the Log Writer use asynchronous commits when writing job log information to the database. Default is OFF.
- PARAM_EVAL_LIMIT_SECS: Maximum time in seconds for parameter evaluation. Default is 5 seconds.
- SET_SSH_EXE_DIR: Identify the location of the ssh executable if you wish to set up jobs that execute scripts on remote servers (see Appendix E for more details on SSH configuration)

# 4 Cronology Console Installation / Upgrade

There is no difference in procedure between installing and upgrading. The procedure is to completely remove any old version of the console and install the latest version.

## 4.1 Prerequisites

### 4.1.1 Installation Variables

During the installation / upgrade you will need to specify the following information:

**Console Application ID**
The APEX application ID to be used for the console installation. Leave as the default 10101 and only change if the install reports a clash of application ids. If a change is required, the application ID must be a positive integer, and cannot be in the reserved range of application IDs (3000 - 8999). It must be less than 3000 or greater than or equal to 9000.

Please have this information to hand **before** you begin the installation.

## 4.2 Installation / Upgrade

The installation / upgrade should be undertaken by a DBA.

### 4.2.1 Copy Installation Media

- logon to the Unix server
- create an installation directory and cd into it
- copy the cronology_4200.zip file into the directory (N.B. the zip file should be FTP'd in **BINARY** mode)
- unzip cronology_4200.zip
- cd into the 'console' directory

### 4.2.2 Create and Edit Installation Variables Script

Rather than prompt the user for input during the installation, a script containing all required installation parameters must be setup before running the installation. This is useful as different parameter set ups can be saved for future use / reference.

- copy the console_install_vars_MASTER.sql script to a name of your choosing. You may choose whatever naming convention you like for the file but typically the name would include the server and database name the install is relevant for
- edit the file and enter all the relevant details – the template gives descriptions and examples for each parameter – save the file

### 4.2.3 Run Installation

To run the installation:

- logon to the target database / PDB via SQL*Plus as the CRONOLOGY user
- run the installation script providing your variable script name as a parameter:

  @console_install <your variable script name>

The installation should complete without any errors. The install will remove any old version of the console if one already exists. If the install fails for any reason you should:

- look to address any database related issues (contact Cronology support if needs be)
- uninstall (see Uninstall section) and re-run the install

If successful the console should be accessible via a web browser. To determine the console URL log in to the APEX admin services page to obtain the **server, port and leading portion of the URL**, this will be determined by your ORDS configuration. The final portion of the URL is: **f?p=CRONOLOGY_CONSOLE**

e.g. **http://<server>:<port>/ords/db1/**f?p=CRONOLOGY_CONSOLE

The logon screen should be presented with your database name displayed (brackets indicates a PDB connection)

## 4.3 Cronology Console Access

Ensure all users have valid database accounts and have been granted **one** of the available Cronology roles: CRONOLOGY_OPERATOR, CRONOLOGY_ADMINISTRATOR or CRONOLOGY_READONLY.

**CRONOLOGY_ADMINISTRATOR**
Has full access to all console functions, i.e. they can create jobs and parameters and manipulate the schedule.

For administrator users, when accessing the console for the first time after installation:

- enter the license key via the About screen
- ensure the messaging options (Main Menu – Settings – Messaging Options) are set correctly, in particular operator e-mail address

**CRONOLOGY_OPERATOR**
Can only view the schedule, execute, cancel and kill jobs. They may not alter the schedule (i.e. cannot create or amend jobs).

**CRONOLOGY_READONLY**
Can only view the schedule and associated job logs. Access is read only, they may not alter the schedule, or execute, cancel or kill jobs. They may not alter the schedule (i.e. cannot create or amend jobs).

You cannot create database accounts via the console - this remains a DBA task.

Existing users with the CRONOLOGY_ADMINISTRATOR role may grant other users access via the Console (Main Menu – Schedule – Manage – Console Access).

## 4.4 Cronology Console Two Factor Authentication

The Cronology Console supports Two Factor Authentication (2FA) by means of Time-based One-Time Passwords (TOTP). Users must have access to a suitable authenticator app on their mobile phone e.g. Google or Microsoft Authenticator. 2FA is OFF by default.

Console 2FA can be set either ON or OFF by using the SET_PARAMETER procedure in the ADMIN package, e.g. to enable 2FA issue the following in SQL*Plus whilst logged on as the CRONOLOGY user:

```
exec admin.set_parameter('CONSOLE_2FA', 'ON');
```

## 4.5 Cronology Console Session Timeout

By default a console session will not timeout / expire. If your organisation requires limits on application sessions then use the SET_PARAMETER procedure in the ADMIN package, e.g. to limit sessions to 24 hours (1440 minutes) then issue the following in SQL*Plus whilst logged on as the CRONOLOGY user:

```
exec admin.set_parameter('CONSOLE_MAX_SESSION_MINS', 1440);
```

Setting a value of 0 will restore the default of no timeout / expiry.

# 5 Uninstall

## 5.1 Server Uninstall

To remove the Cronology Server installation simply run the uninstall.sql script supplied with the server code whilst connected as a **DBA user or SYS** to the database / PDB - this drops:

- the CRONOLOGY schema and all its associated objects
- the CRONOLOGY_READONLY, CRONOLOGY_OPERATOR and CRONOLOGY_ADMINISTRATOR roles
- the CRONOLOGY network access controls

**THIS COMPLETELY UNINSTALLS THE CRONOLOGY SERVER INSTALLATION (IT DOES NOT ROLLBACK AN UPGRADE).**

## 5.2 Console Uninstall

To remove the Cronology Console APEX application simply run the console_uninstall.sql script supplied with the console code as a **DBA user or SYS** whilst connected to the database / PDB - this drops:

- the Cronology Console application
- the CRONOLOGY APEX workspace

# 6 General Considerations

## 6.1 Changing the CRONOLOGY schema password

To change the CRONOLOGY password a DBA should use the CRONOLOGY.ADMIN.SET_SERVER_PASSWORD procedure.

**This changes the password for the CRONOLOGY schema and also sets the new password (encrypted) in the system parameters table. It is imperative the schema password and system parameter entry are kept in sync, i.e. always use this procedure and do not issue an ALTER USER CRONOLOGY IDENTIFDIED BY <password> command stand alone.**

## 6.2 Server Overwrite / Restore / Copy - License Keys

Keep a log of your supplied license keys and to which database names they belong. If you are overwriting / restoring an installation from another database and the database names differ – you will be prompted to enter the valid license key the first time the console connects to the restored installation.

If you are creating a new database from a copy and you do not already have a key for the new database name you will be required to obtain a key from Cronology Software Ltd.

# Appendix A - Cronology ADMIN Package

The ADMIN package provided with this release allows a DBA to perform useful administrative tasks. **All procedures that update data implicitly perform a commit.** Procedures in the package are as follows:

- **PROCEDURE CONSOLE_2FA_BYPASS**
  Will bypass two factor authentication for the specified user. Can be used if, for example, the user has lost access to their authenticator app.

- **PROCEDURE CONSOLE_2FA_RESET**
  Will cause the console to ask the user to set up two factor authentication again, allowing them to re-scan QR code in their authenticator app.

- **PROCEDURE DELETE_AUDIT_HISTORY**
  Deletes schedule audit history retaining the specified number of months of data. By default the auditing of schedule settings are not deleted. This can be overridden to purge ALL audit history.

- **PROCEDURE DISABLE_ALL_MESSAGING**
  Sets all messaging options off. This can also be done via the console. Optionally can reset all messaging configuration: email server details, socket settings, message templates, operator email address, bespoke messaging procedure and all job specific email lists (completion and failure). This procedure is useful when an environment has been copied to another database and you wish to remove all messaging configuration to avoid messages being sent from the new environment.

- **FUNCTION GET_PARAMETER RETURNS VARCHAR2**
  Returns the supplied system parameter value (entries in CRONOLOGY.PARAMETERS table with a PARAM_ID < 0) as seen by the Cronology Server and Console sessions

- **FUNCTION GET_RUNTIME_CREDENTIALS RETURNS VARCHAR2**
  Utility function to display the current runtime credentials in effect.

- **PROCEDURE MARK_JOB_AS_KILLED**
  Forces a jobs status to KILLED (last resort if cannot kill via console / OS i.e. process id not captured)

- **PROCEDURE PURGE_HISTORY**
  Purges **all** historical job information but leaves the schedule (jobs, parameters, dependencies etc.) intact.

- **PROCEDURE PURGE_SCHEDULE**
  Purges the entire schedule – all parameter and job information is purged along with all historical job information.

- **PROCEDURE REBUILD_JOB_LOGS**
  Rebuilds the JOB_LOG table as either a partitioned table or non-partitioned table.

- **PROCEDURE REGISTER_JSSU_CREDENTIALS**
  Must be used in conjunction with sys.dbms_java.set_runtime_exec_credentials to correctly set JSSU runtime credentials. Please contact support@cronology.co.uk before using this procedure.

- **PROCEDURE RELOAD_JAVA_LIB**
  If for any reason the Cronology JAVA library becomes corrupted or requires reloading, this procedure can be used.

- **PROCEDURE RESET_HOST_NAME**
  Resets the HOST_NAME parameter to the host name running the database instance running Cronology.

- **PROCEDURE RESET_JOB_SEQUENCE**
  Resets the sequence used for jobs. Only use this if you have removed the entire schedule (including offline jobs).

- **PROCEDURE RESET_PARAMETER_SEQUENCE**
  Resets the sequence used for job parameters. Only use this procedure if you have removed all job parameters.

- **PROCEDURE RESUME_SCHEDULE**
  If the schedule has been suspended (SUSPEND_SCHEDULE), resume job executions.

- **PROCEDURE SET_MESSAGE_TEMPLATE**
  Use this procedure to define a template / format for messages used for either the bespoke, socket or SNMP messaging. A default template may be set for all messages or individual templates can be set for each message type. **Templates may also be set via the Cronology Console (Messaging Options).** See the MESSAGE_TEMPLATES table for available templates. When using templates Cronology uses the following precedence:

  1) message type specific template (e.g. SOCKET_TEMPLATE_KILLED for job kill notifications via socket)
  2) default template (SOCKET_TEMPLATE_DEFAULT, BESPOKE_MESSAGE_DEFAULT or SNMP_MESSAGE_DEFAULT)
  3) standard message (no templates defined i.e. all templates are set to <NONE>) – only BESPOKE and SOCKET messaging will send a standard message, SNMP messaging MUST have templates defined. N.B. E-mail messaging does not use templates. E-mail messages will always use the 'standard' Cronology message.

  For example, to set the default socket template:

  exec admin.set_message_template('SOCKET_TEMPLATE_DEFAULT','Job: #JOB_NAME# Status: #STATUS# Timestamp: #SYSDATE_DD-MON-YYYY HH24:MI:SS#')

  SNMP trap messaging requires the template to contain all the command line options that would normally be passed to the snmptrap (snmp_trapsnd for Tru64) UNIX executable, e.g.

  For example, in its most basic form, a trap could be invoked with the following options:

```
-v 1 -ci <SNMP_Server_Host> <Sender_OID> <Sender_Host> <Trap_Type>
<Specific_Type> '' <OID> s "<Content1>" <OID> s "<Content2>"
```

  So the template could be set (with greatly simplified OIDs):

```
exec admin.set_message_template('SNMP_TEMPLATE_DEFAULT',q'!-v 1 -ci
192.168.1.21 1.2.3.4 #HOST_NAME# 6 1 '' 1.1 s "Job: #JOB_NAME# Status: #STATUS#
Timestamp: #SYSDATE_DD-MON-YYYY HH24:MI:SS#" 1.2 s "#JOB_PRIORITY#"!');
```

When defining templates the following substitution variables may be used (the # characters must be included in the template for the substitution to take place):

| | |
|---|---|
| #JOB_NAME# | The job name that triggered the message |
| #STATUS# | The status of the triggering event, status will be either:<br>'KILLED', 'OVERDUE', 'CANCELLED', 'FAILED', 'WAITING', 'RESOLVED', 'RUNTIME' (job has exceeded runtime warning threshold), 'INTERNAL' (internal Cronology error) |
| #SUBJECT# | The default / standard Cronology message |
| #INSTANCE_NAME# | The Oracle instance name Cronology is running on |
| #HOST_NAME# | The host name Cronology is running on |
| #INFO# | Additional information Cronology may provide with the message |
| #SYSDATE_<DATE FORMAT># | The current date and time, please provide a valid Oracle date format e.g. #SYSDATE_DD-MON-YYYY HH24:MI:SS# |
| #USER_NAME# | The database user who initiated the message (only really useful for job kill, cancellation and resolved messages) |
| #JOB_PRIORITY# | The priority of the job that triggered the message |
| #JOB_MESSAGE_TEXT# | The supplemental messaging information defined for the job |

To reset / clear a template call the procedure with the template name and no / a null template (template gets set to <NONE>). You may provide the Oracle wildcard (%) character in the template name parameter if you wish to update multiple templates at once.

- **PROCEDURE SET_PARAMETER**
  Sets the supplied system parameter in the CRONOLOGY.PARAMETERS table.

- **PROCEDURE SET_SERVER_PASSWORD**
  This procedure sets an encrypted version of the CRONOLOGY schema password in the PARAMETERS table for use by the Cronology background processes (Job Poller, Log Writer and Message Server). It can also optionally set the password for the CRONOLOGY schema itself to save having to issue a separate ALTER USER command. Please ensure the PARAMETERS table entry is in sync with the database password. For 11g onwards – if the CRONOLOGY schema approaches password expiry the Cronology application will attempt to automatically reset it to its existing value. This is done to avoid the background processes failing / generating ORA-28002 (Password will expire in N days) errors**.** This effectively means the password will never expire, assuming of course immediate password reuse is allowed. **If required, use this procedure to change the password at regular intervals as required by your organization.**

- **PROCEDURE SET_SMTP_SETTINGS**
  If e-mail messaging is required, use this procedure to set the SMTP server details. Run with no parameters to "reset" (details set to <NONE>). To determine your server name / server IP address issue the following UNIX command:

  echo test | mailx -v <your_email_address>

  You should see a line similar to: ... Connecting to **<SMTP server name/ IP address>** via relay ...

  Port 25 is most common / the default. If username and password authentication are not required by the SMTP server please leave the sender e-mail address, username and password parameters null. If authentication is required, in most cases the SMTP sender e-mail address should be the same as the SMTP username.

- **PROCEDURE SET_SNMP_EXE_DIR**
  If SNMP trap messaging is required, use this procedure to set the location of the snmptrap (snmp_trapsnd for Tru64) executable. On Linux, Solaris and HP-UX use the following command to determine the correct directory:

  unalias -a ; dirname `which snmptrap`

  On Tru64 use:

  unalias -a ; dirname `which snmp_trapsnd`

  Use the SET_MESSAGE_TEMPLATE procedure to set the command line parameters for the SNMP traps.

- **PROCEDURE SET_SOCKET_SETTINGS**
  If socket messaging is required, use this procedure to set the socket server details. Run with no parameters to "reset" (details set to <NONE>)

- **PROCEDURE SET_SSH_EXE_DIR**
  If jobs are to be run over SSH, use this procedure to set the location of the SSH executable, use the following command to determine the correct directory:

  unalias -a ; dirname `which ssh`

- **PROCEDURE SET_SUDO_CREDENTIALS**
  Used to set SUDO credentials for runtime execution. Please contact support@cronology.co.uk before using this procedure.

- **PROCEDURE SET_SUDO_EXE_DIR**
  Sets the location of the SUDO executable for your environment.

- **PROCEDURE START_ALL**
  Starts all Cronology Server processes (Job Poller, Log Writer and Message Server)

- **PROCEDURE START_JOB_POLLER**
  Starts the Cronology Job Poller

- **PROCEDURE START_LOG_WRITER**
  Starts the Cronology Log Writer

- **PROCEDURE START_MESSAGE_SERVER**
  Starts the Cronology Message Server

- **PROCEDURE STOP_ALL**
  Stops all Cronology Server processes (Job Poller, Log Writer and Message Server)

- **PROCEDURE STOP_JOB_POLLER**
  Stops the Cronology Job Poller immediately or, if a stop date and time is provided as an optional parameter the job poller will stop at the specified date and time.

- **PROCEDURE STOP_JOB_POLLER_CANCEL_DELAYED**
  If a delayed shutdown has been requested calling this procedure cancels the delayed shutdown. N.B. It only **cancels** a delayed shutdown, it does not shutdown the job poller.

- **PROCEDURE STOP_LOG_WRITER**
  Stops the Cronology Log Writer

- **PROCEDURE STOP_MESSAGE_SERVER**
  Stops the Cronology Message Server

- **PROCEDURE SUSEPEND_SCHEDULE**
  Suspends the schedule, running jobs will complete but no further jobs are launched. Use RESUME_SCHEDULE to resume normal operations.

- **FUNCTION SYSTEM_INFO RETURNS CLOB**
  Returns Cronology system information e.g. Oracle and UNIX OS details, Cronology version and current system parameters.

- **PROCEDURE TEST_OS_CMDS**
  Tests the environment is configured correctly in order to run OS commands. The procedure will complete successfully if the environment is ok, or will raise an appropriate exception.

- **PROCEDURE UPD_SCHEDULE_CONNECT_STRING**
  Will update all jobs in the schedule, changing an old connect string to a new connect string. This procedure converts connect strings to upper case.

- **PROCEDURE UPD_SCHEDULE_PASSWORD**
  Will update all jobs in the schedule updating a password for a given username and connect string. If no connect string is defined for the jobs then supply NULL as the connect string parameter. This procedure converts connect strings and usernames to upper case.

- **PROCEDURE UPD_SCHEDULE_SSH_CONNECTION**
  Will update all jobs in the schedule updating one SSH connection to another.

- **PROCEDURE WAIT_FOR_RUNNING_JOBS**
  Use for scripting, will wait for any running jobs to complete. Useful when used in conjunction with SUSPEND_SCHEDULE.

# Appendix B - Cronology API Package

The API package has execute privilege granted to PUBLIC and provides a number of procedures that application developers may use to interface with the Cronology Server processes:

- **PROCEDURE LOG_LINE**
  This procedure may be called from within a PL/SQL program unit to allow real-time output to the job log when the job is run via Cronology. Traditionally DBMS_OUTPUT would be used to produce output from within PL/SQL, this however waits until the program unit has completed before producing the output. Programmers may use CRONOLOGY.API.LOG_LINE where they would normally use DBMS_OUTPUT.PUT_LINE. If the program unit is run outside of Cronology then LOG_LINE acts exactly the same as DBMS_OUTPUT.PUT_LINE. An option parameter (P_MODE) exists for the procedure. If LOG_LINE is called with P_MODE = 'A' then the line will be appended to previous line.

- **PROCEDURE EXECUTE_JOB**
  This procedure will execute a job in the Cronology schedule. This procedure is overloaded to take either a job name (CRONOLOGY.JOB table JOB_NAME column) or the job id (CRONOLOGY.JOB table JOB_ID column – using job ids is not recommended other than for development or testing purposes as job ids are unlikely to be consistent across your environments). Other parameters: P_RSA = run stand alone (Y or N), P_UOP = use override parameters (Y or N), P_FAIL_ON_ERROR = if the job fails raise an exception (Y or N). This procedure allows jobs to be executed programmatically via Cronology from within PL/SQL program units if needs be.

- **PROCEDURE DELETE_HISTORY**
  A procedure named DELETE_HISTORY has been provided in the CRONOLOGY.API package to delete historical job information. This is a recommended Cronology housekeeping routine, as such a job should be created in the schedule to call the CRONOLOGY.API.DELETE_HISTORY procedure at an interval and time that suits resource availability on your system. The procedure deletes data based on the JOB_HISTORY_RETENTION parameter, set via the Console under Main Menu -> Schedule -> Settings -> Job History Retention). **The CRONOLOGY.API package has execute privilege granted to PUBLIC, but internal validation inside the DELETE_HISTORY procedure ensures that only DB users granted the DBA, CRONOLOGY_ADMINISTRATOR or CRONOLOGY_OPERATOR role may execute it. Optional parameters are available for this procedure:**

  | | | |
  |---|---|---|
  | **P_SHRINK_SPACE** | **BOOLEAN** | **defaults to FALSE** |
  | **P_COMMIT_BATCH_SIZE** | **NUMBER** | **defaults to 10000** |
  | **P_MAX_DELETE_TIME_HRS** | **NUMBER** | **defaults to NULL** |

  **Shrink space set to TRUE will attempt to compact the space used by the JOB_LOGS table.**
  **Commit batch size will delete job run history rows in batches of this size and commit after each batch. This may be useful if the DELETE_HISTORY procedure is being invoked for the first time and a lot of historical data is to be deleted.**
  **Max delete time can be set to limit the run time of the procedure. The procedure will exit once this run time has been reached regardless of how much data has been deleted. If a lot of data needs to be deleted this may be useful to spread the load over multiple executions.**

Other procedures exist in the API package are not intended for general use, they are primarily used for supporting the Cronology processes or for use when instructed by Cronology support.

# Appendix C - Disabling Job Poller Autostart

If a database has been shut down (with Autostart left ON) and the DBA does not wish the Cronology Job Poller to automatically start (and hence possibly launch jobs) when the database starts up, one of the following two methods may be employed:

## Method 1

- startup mount;
- execute dbms_application_info.set_module('CRONOLOGY','DISABLE_AUTOSTART');
- alter database open;

The Job Poller will detect this setting when starting up and shut itself down - it will also set the JOB_POLLER_AUTO_START system parameter to 'OFF' so that subsequent database startups will not start the Job Poller. Autostart can be turned on again via the Cronology Console.

## Method 2

- startup mount;
- alter system set "_system_trig_enabled" = FALSE;
- alter database open;

This method will disable the trigger that starts the Job Poller. Depending on how this parameter is set (memory only or permanently in the database pfile / spfile) Autostart should be turned off via the Cronology Console to avoid future Autostarts when the database is started.

**WARNING: ALL** database startup triggers will be disabled if this method is used.

# Appendix D - Resetting an Expired License Key

If a license key has been allowed to expire then it can no longer be updated via the Cronology Console.

A valid key must be obtained from Cronology Software Ltd. A DBA should then use the following PL/SQL to update the license key:

```
declare
   lic_key varchar2(64) := '&LICENSE_KEY';
begin
   begin
    -- If license has expired the package initialization sections will raise an error, ignore it ...
       cronology.admin.set_parameter('LICENSE_KEY_PART_1',lic_key);
   exception
      when others then
        null;
   end;
   cronology.admin.set_parameter('LICENSE_KEY_PART_1',lic_key);
end;
/
```

# Appendix E - Configuring Passwordless SSH

If you require remote jobs to run over SSH you will need to configure passwordless SSH connectivity between your database server and the remote servers. The basic steps are as follows:

- **Generate an SSH key**
  Logon to the database server using **the same runtime credentials that Cronology runs jobs as** – by default this is typically an account called 'oracle'. Generate a key by issuing the following command:

  ssh-keygen –t rsa

  Accept the default file location to save the key in and do not enter a passphrase (hit return). This will generate a key in the **id_rsa.pub** file located in **$HOME/.ssh**

- **Distribute key**
  Copy the key into the $HOME/.ssh/authorized_keys file of all users you wish to connect to on all required servers**. It is vital that the .ssh directory and authorized_keys file have UNIX permissions 700 (rwx------).**

Hostnames on your network may be used or IP addresses, in the following example IP addresses are used:

Database is running on 192.168.65.102 under 'oracle' account:

```
[oracle@192.168.65.102]$ whoami
oracle

[oracle@192.168.65.102]$ pwd
/home/oracle

[oracle@192.168.65.102]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/oracle/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/oracle/.ssh/id_rsa.
Your public key has been saved in /home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:
15:3c:1c:fd:29:3d:67:61:50:5a:70:e1:2d:e7:59:35 oracle@localhost.localdomain
```

To see the key generated:

```
[oracle@192.168.65.102]$ cd .ssh

[oracle@192.168.65.102]$ ls -ltr
total 8
-rw-r--r-- 1 oracle oinstall  410 Apr 15 13:18 id_rsa.pub
-rw------- 1 oracle oinstall 1675 Apr 15 13:18 id_rsa

[oracle@192.168.65.102]$ cat id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAz7lGeU5PzdAC1/J18FM7/z8gPVlqYYt/DBHqoru5fQ3CiWJO51hCOSMEUWUYbZQ+2Az
1sFwIiU5dA8+/TYWsinF2G+RBbXl0CALx5H/asZo1BwRKOXgAPVYJUBEU0jgQQo/xwbD3LM+Fv60WqhlO2i7rpg8BJsNGBx
pgD5wIEMw0ZSDKOHKycAWLV9DdMACVSSMByG0Ngss0zNIjlGyXxQSuig7g53X5b8LMAG8mPLCkxxnGZDNkDxGaHI21uelSE
NM7JtFZmbfYUUpawRhbBJ9tA5k26UQR0xMFUSC+pVDwZcPwI6lXtNjpIuh95CQUt+rjn55uceCzCw3PO1RH7w==
oracle@localhost.localdomain
```

Say we need to connect as user 'fred' on 192.168.56.103 to execute a script. On this server, logged on as 'fred' paste the whole key into the authorized_keys file under $HOME/.ssh :

```
[fred@192.168.56.103]$ whoami
fred

[fred@192.168.56.103]$ pwd
/home/fred

[fred@192.168.56.103]$ ls -ltra
total 44
drwxr-xr-x 4 fred fred 4096 Apr 15 13:45 .mozilla
-rw-r--r-- 1 fred fred  515 Apr 15 13:45 .emacs
-rw-r--r-- 1 fred fred  124 Apr 15 13:45 .bashrc
-rw-r--r-- 1 fred fred   33 Apr 15 13:45 .bash_logout
drwxr-xr-x 5 root root 4096 Apr 15 13:45 ..
-rw-r--r-- 1 fred fred  282 Apr 15 13:46 .bash_profile
-rw------- 1 fred fred 2209 Apr 15 13:53 .viminfo
drwx------ 2 fred fred 4096 Apr 15 13:53 .ssh
-rw------- 1 fred fred  213 Apr 15 13:54 .bash_history
drwx------ 4 fred fred 4096 Apr 15 13:54 .

[fred@192.168.56.103]$ cd .ssh

[fred@192.168.56.103]$ cat authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAy5EE5Oo1gaMbqTL3uQKjATunxgPhITtEscku0CYKUm5Tp5Edy/za4j1tYVNfl5G41eU
JnmqH8iJPYFs8Mrn3G3fox0zkn+Hi1EGKgg/GhE7ZId+AYnm4liX4mLAb5WvqiprHdSGsM3ST+yQRqpIsJ1BUNEJwfJ4ZX2
AorPRJy9yZvttrNilrczefAhg3LAagv1bInlTGM2rGgMdeyAKWw4EuX+9+7Po5PORrRCQmfB0fqkcT510wt1iUeu2EiM5fn
XYmecu4Qy/3WRBwmx7ZF+O94bOe9Z1Fjgu/5VlpVl1UIG53zX41mzGNhQmPIBySymyQ2DsoAskZKOdNaWW4Ew==
oracle@localhost.localdomain
```

**It is vital that the .ssh directory and authorized_keys file have 700 permissions, do this with:**

```
[fred@192.168.56.103]$ cd $HOME
[fred@192.168.56.103]$ chmod 700 .ssh
[fred@192.168.56.103]$ chmod 700 .ssh/authorized_keys
```

We can test the setup by running some commands over ssh from our database server:

```
[oracle@192.168.56.102]$  ssh fred@192.168.56.103 'whoami ; /sbin/ifconfig eth0'
fred
eth0      Link encap:Ethernet  HWaddr 08:00:27:81:27:B6
          inet addr:192.168.56.103  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe81:27b6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10262 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5344 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1095473 (1.0 MiB)  TX bytes:931333 (909.5 KiB)
          Base address:0xd010 Memory:f0000000-f0020000
```

# Appendix F - Installing APEX

This appendix provides a minimalistic install guide for installing APEX into your Oracle database.

| Action | Command |
|---|---|
| Logon to your DB server as the oracle user and switch to a suitable install location | `cd /home/oracle` |
| Create an install directory | `mkdir apex192` |
| | `cd apex192` |
| Download the APEX 19.2 media (ENG only is sufficient) and the latest patch bundle https://www.oracle.com/tools/downloads/apex-192-downloads.html and copy into the apex192 directory | |
| Unzip the media (inflates to two directories: apex and 30392181) | `unzip apex_19.2_en.zip`<br>`unzip p30392181_1920_Generic.zip` |
| Switch to apex directory to install APEX | `cd apex` |
| Connect to the database / PDB as SYS | `sqlplus sys/<pw> as sysdba` |
| Create a tablespace to hold APEX objects eg | `-- For Oracle Managed Files (OMF)`<br>`CREATE TABLESPACE apex DATAFILE SIZE 100M AUTOEXTEND ON NEXT 1M;`<br>`-- For non-OMF`<br>`CREATE TABLESPACE apex DATAFILE '/opt/oracle/oradata/ORCLCDB/ORCLPDB1/apex01.dbf' SIZE 100M AUTOEXTEND ON NEXT 1M;` |
| Install APEX (this may take several minutes) | `@apexins.sql APEX APEX TEMP /i/` |
| Create the ADMIN user for APEX | `@apxchpwd.sql` |
| Open the APEX_PUBLIC_USER account | `ALTER USER APEX_PUBLIC_USER IDENTIFIED BY "<pw>" ACCOUNT UNLOCK;` |

**If using ORDS:**

| | |
|---|---|
| Create the APEX_LISTENER and APEX_REST_PUBLIC_USER users (enter passwords when prompted) | `@apex_rest_config.sql` |

**If using EPG:**

| | |
|---|---|
| Connect to the database or **MAIN CONTAINER of a CDB** | `sqlplus sys/<pw> as sysdba` |
| Configure EPG specifying the directory ABOVE the unzipped apex directory eg | `@apex_epg_config.sql` **`/home/oracle/apex192`** |
| Set SHARED_SERVERS to a minimum of 5 for performance. If already 5 or above then skip this step | `alter system set shared_servers=5 scope=both;` |
| Open the ANONYMOUS account | `ALTER USER anonymous IDENTIFIED BY <pw> ACCOUNT UNLOCK CONTAINER=ALL;`<br>`-- Or if non-CDB`<br>`ALTER USER anonymous IDENTIFIED BY <pw> ACCOUNT UNLOCK;` |
| Connect to the database / PDB as SYS | `sqlplus sys/<pw> as sysdba` |
| Find and set the EPG port to listen on, eg | `SELECT DBMS_XDB.gethttpport FROM DUAL;`<br>`-- If value is 0, set and enable the port with`<br>`EXEC DBMS_XDB.sethttpport(8080);` |
| Check you can access APEX via a browser using a URL like http://machine:8080/apex | |

**Apply latest APEX Patch**

| | |
|---|---|
| Switch to patch directory | `cd 30392181` |
| If using ORDS, stop ORDS | |
| Connect to the database / PDB as SYS | `sqlplus sys/<pw> as sysdba` |
| Install the patch (this may take a few minutes) | `@catpatch.sql` |
| Check the patch is installed. This command should return the date time of the patch install | `select APEX_INSTANCE_ADMIN.GET_PARAMETER ('APEX_19_2_0_PATCH_30392181') from dual;` |

# Appendix G - Installing ORDS (DB server)

This appendix provides a minimalistic install guide for installing ORDS onto your DB server. You will need a working installation of Java on the DB server. This guide is based on Java8 and ORDS v20.3.0 running over HTTP using the "standalone" built in Jetty web server.

**If Java is not available / installed** (test with java –version) install it:

| Action | Command |
|---|---|
| Logon to your DB server as the oracle user and switch to a suitable install location | `cd /usr` |
| Create a java directory | `mkdir java` |
| | `cd java` |
| Download the Java JRE relevant to your platform eg https://java.com/en/download/help/linux_x64_install.html#download and copy the .gz file into the java directory | |
| Unzip the media eg | `tar zxvf jre-8u333-linux-x64.tar.gz` |
| Add java to your path | `vi ~/.bashrc and add`<br>`export PATH=$PATH:/usr/java/jre1.8.0_333/bin` |
| Logoff and logon agan, check java is available | `java –version`<br><br>`The java version should be shown eg`<br><br>`java version "1.8.0_333"`<br>`Java(TM) SE Runtime Environment (build 1.8.0_333-b02)`<br>`Java HotSpot(TM) 64-Bit Server VM (build 25.333-b02, mixed mode)` |

**Install ORDS**

| | |
|---|---|
| Logon to your DB server as the oracle user and switch to a suitable install location | `cd /home/oracle` |
| Create an ords directory | `mkdir ords` |
| | `cd ords` |
| Download ORDS https://www.oracle.com/database/technologies/appdev/rest-data-services-downloads.html and copy the .zip file into the ords directory | |
| Unzip the media eg | `unzip ords-20.3.0.301.1819.zip` |
| Make a config directory | `mkdir conf` |
| Tell ORDS to use the conf directory | `java -jar ords.war configdir /home/oracle/ords/conf` |
| Set required properties before configuring ORDS | `vi /home/oracle/ords/params/ords_params.properties`<br><br>`set the standalone port accordingly eg`<br><br>`standalone.http.port=8081`<br><br>`Add two entries:`<br><br>`standalone.mode=true`<br>`standalone.static.path=`<br><br>`Save the file` |
| Configure ORDS | `java -jar ords.war` |

Follow configuration prompts as required. To point ORDS at the local DB accept the default server, port and enter the database service name. Enter passwords as required. If you are required to enter an “admin” user to create DB users – use SYS. To run APEX only you do not need to enable any extra features.

Sample:

```
java -jar ords.war

Enter the name of the database server [localhost]:
Enter the database listen port [1521]:
Enter 1 to specify the database service name, or 2 to specify the database SID [1]:
Enter the database service name:orclpdb1
Enter the database password for ORDS_PUBLIC_USER:
Confirm password:

Retrieving information.
Enter 1 if you want to use PL/SQL Gateway or 2 to skip this step.
If using Oracle Application Express or migrating from mod_plsql then you must enter 1 [1]:
Enter the database password for APEX_PUBLIC_USER:
Confirm password:
Enter 1 to specify passwords for Application Express RESTful Services database users (APEX_LISTENER,
APEX_REST_PUBLIC_USER) or 2 to skip this step [1]:
Enter the database password for APEX_LISTENER:
Confirm password:
Enter the database password for APEX_REST_PUBLIC_USER:
Confirm password:
Enter a number to select a feature to enable:
   [1] SQL Developer Web  (Enables all features)
   [2] REST Enabled SQL
   [3] Database API
   [4] REST Enabled SQL and Database API
   [5] None
Choose [1]:5
2022-05-31T19:21:30.474Z INFO        reloaded pools: []
2022-05-31T19:21:30.477Z INFO          Oracle REST Data Services schema version 20.3.0.r3011819 is
installed.
Enter the APEX static resources location:
Enter 1 if using HTTP or 2 if using HTTPS [1]:1
```

ORDS then starts. Ctrl-C to cancel.

**N.B. You do not need to specify or copy static resource files to run Cronology. This may not be the case for other APEX applications.**

You will probably need to write a script to start ORDS In the background and add to your server startup files eg to start ORDS in standalone mode:

```
java -jar ords.war standalone &
```

Useful links:

https://oracle-base.com/articles/misc/oracle-rest-data-services-ords-standalone-mode