

CRONOLOGY



• JOB SCHEDULING SOFTWARE ON ORACLE •

VERSION 4.1

CONSOLE USER GUIDE

© CRONOLOGY LIMITED
WWW.CRONOLOGY.CO.UK

1	Introduction	3
2	Supported Environments.....	3
3	Required Reading	3
4	Getting Started.....	4
4.1	User Types.....	4
4.2	Console Screen Layout	4
4.3	Navigation and Default Actions	7
4.4	Run History.....	9
4.5	Job Log	9
4.6	Menu Structure.....	10
5	A Quick Example For Administrators	11
6	Parameter Maintenance.....	13
6.1	Creating a Parameter	13
6.2	Amending a Parameter	14
6.3	Deleting a Parameter	14
7	Job Maintenance	15
7.1	Creating a Job	15
7.2	Amending a Job	19
7.3	Deleting a Job	19
7.4	Killing a Job.....	19
7.5	Cancelling a Job	19
7.6	Taking a Job Offline.....	20
7.7	Exporting a Job	20
7.8	Managing Job Dependencies	20
8	Executing Jobs Manually.....	22
9	Schedule Maintenance.....	23
9.1	Change Control	23
9.2	Suspend the Schedule	23
9.3	Advance the Schedule	23
9.4	Remove E-mail Recipients	23
9.5	Exporting the Schedule	24
10	Monitoring Database Sessions	25
10.1	Username Menu	27
10.2	Kill Exclusions	27
10.3	Tracing Sessions.....	27
11	Job Analysis	29
11.1	Activity Query.....	29
11.2	Status By Occurrence	29
11.3	Status By Percentage	29
11.4	Average Durations	29
12	Settings	30
12.1	Messaging Options.....	30
12.2	Console Settings	31
12.3	Advanced	31
12.4	Message Templates	32
13	Job Poller.....	33
14	Log Writer	33
15	Message Server	33

1 INTRODUCTION

The Cronology Console is the GUI front end for the Cronology product. It allows operators and administrators to control, examine, set up and execute jobs.

The console provides a consolidated view of your entire job schedule and provides the ability to view job logs, job source code, monitor database sessions, view active SQL / explain plans and much more.

A key feature of the console is that all job types (e.g. SQL script, stored procedure, shell script, JAVA executable) are all administered the same way – this can be advantageous during the testing phase of a project when testers may need to run many jobs on an ad-hoc basis, view the output etc without having to worry about how to invoke the underlying source code for each job.

The console is written in Oracle Forms but behaves more like an MS Windows client application i.e. traditional Forms keys like F8 to execute a query are disabled – all actions are driven by menu selections, button presses etc.

2 SUPPORTED ENVIRONMENTS

In order to run the console you will require the Oracle Forms Runtime 6i for Windows installed on your PC. Web based forms 6i is not currently supported. **Version 6.0.8.24.1 (Patch 15) or above is required for the console to operate correctly.** Your display must be at least 1024 x 768. **Forms Runtime 6i for Windows will NOT run against a database with an AL32UTF8 character set**

3 REQUIRED READING

If you have not already done so, please read the Cronology 4.1 Overview document. The key concepts discussed there are assumed knowledge when using this guide.

4 GETTING STARTED

4.1 User Types

Console access is granted via database roles. Database users can be assigned one of three different user types in order to access the Console:

- **Administrator** - access to all menu options and can set up and alter the job schedule i.e. create and amend jobs, parameters, dependencies etc. Administrators may also grant Console access to other database users via the console
- **Operator** - access to most menu options, cannot change the job schedule but may run jobs manually
- **Read Only** - access to most menu options but can view schedule information only.

Although Operator and Read Only users may have access to certain menus – the screens may only be available in read only mode (i.e. update or save buttons may be disabled).

4.2 Console Screen Layout

The console main screen is as follows

The screenshot displays the Cronology Console interface. At the top, it shows the connection details: 'Connected To: OELS_ELEVENG' and 'Username: PV'. The 'Status Summary' indicates '1 FAILED'. Below this is a table of jobs with columns for Job Name, Status, Start Dttm, End Dttm, Duration, Frequency, and RSA UOP. The 'cleanup' job is highlighted in red, indicating it is 'FAILED'.

Job Name	Status	Start Dttm	End Dttm	Duration	Frequency	RSA UOP
TEST 1	COMPLETE	09-JUN-2010 21:01:44	09-JUN-2010 21:01:49	00:00:00:05	5 MINS	
• test 2	COMPLETE	09-JUN-2010 21:01:51	09-JUN-2010 21:01:56	00:00:00:05	5 MINS	
• test 3	COMPLETE	09-JUN-2010 21:01:56	09-JUN-2010 21:02:01	00:00:00:05	5 MINS	
• test 4	COMPLETE	09-JUN-2010 21:02:01	09-JUN-2010 21:02:06	00:00:00:05	5 MINS	
• test 5	COMPLETE	09-JUN-2010 21:01:56	09-JUN-2010 21:02:01	00:00:00:05	5 MINS	
test 6	WAITING			00:00:06:18	10 MINS	
password change	FAILED	09-JUN-2010 21:06:12	09-JUN-2010 21:06:15	00:00:00:03	CRON	
api test	RUNNING	09-JUN-2010 21:03:43		00:00:05:43	CRON	
placeholder test					MONTHLY	
cleanup					WEEKLY	
2					WEEKLY	
shell test					MONTHLY	

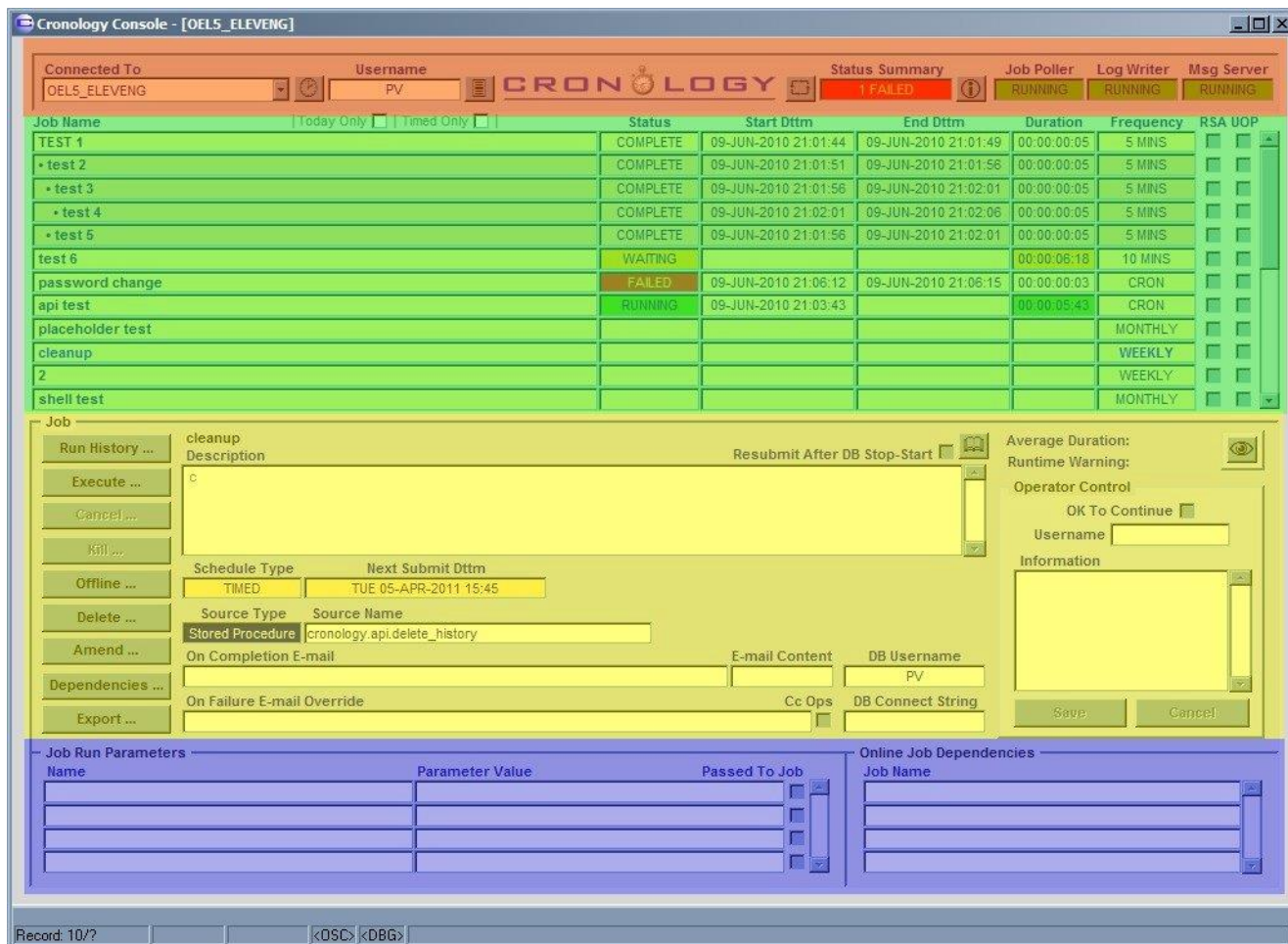
The 'cleanup' job details are shown below the table:

- Job:** cleanup
- Description:** c
- Schedule Type:** TIMED
- Next Submit Dttm:** TUE 05-APR-2011 15:45
- Source Type:** Stored Procedure
- Source Name:** cronology.api.delete_history
- DB Username:** PV
- DB Connect String:**





Additional controls include 'Run History ...', 'Execute ...', 'Cancel ...', 'Kill ...', 'Offline ...', 'Delete ...', 'Amend ...', 'Dependencies ...', 'Export ...', 'Average Duration', 'Runtime Warning', 'Operator Control', and 'Information'.

CRONOLOGY
 - VERSION 4.1 -
 - CONSOLE USER GUIDE -

The main screen is split into four distinct areas:



1) **Header / summary information** (orange): showing the current database connection, user, schedule summary and server process statuses. This section also contains four buttons:

Button	Keyboard Shortcut	Description
	F7	Database server date and time N.B. this may be different to the date and time on your PC
	F6	Main Menu The menu structure is described in detail at the end of this section
	F5	Refresh Display Forces a refresh of the current console display
	F1	Information Displays details of the Cronology installation, runtime environment and license information. License Keys can be updated via this screen

To quickly reconnect to a previous database, click on the Connected To dropdown and select the database from the list.

Double Click Functionality:

- **Username:** a double click will display information about all Cronology Console users on this database
- **Status Summary:** a double click will filter the job display to only show jobs with a status matching the status summary
- **Job Poller:** a double click will display the history (start up, shutdowns etc of the server process)
- **Log Writer:** a double click will display the history (start up, shutdowns etc of the server process)
- **Message Server:** a double click will display the history (start up, shutdowns etc of the server process)

2) Job Schedule (green): the job list is ordered such that jobs that are due to run next appear at the top of the display. As jobs run and complete this view changes dynamically. The currently selected job appears in blue bold font.

Checkbox Filters:

- **Today Only:** shows only timed jobs and their children that are scheduled to run today
- **Timed Only:** shows only timed jobs (and not any children)



Right Click Popup Menus:

- **Job Name:** by right clicking on the Job Name field a popup menu of actions becomes available. Most of these are available by clicking the control buttons in the next section (i.e. this provides another / more convenient way to access job functions)
- **Status:** by right clicking on the status field you can filter the display by status, this can be useful to, for example, show all FAILED jobs

Double Click Functionality:

- **Job Name:** a double click will display the latest / current job log for the job

3) Job Detail (yellow): shows detail on the currently selected job. This section also includes two buttons:

Button	Keyboard Shortcut	Description
	<None>	Documentation Link Click this button to view any documentation set up for the job
	F2 - find by Name F3 - find next F4 - find by Source	Find Job Use this button / shortcuts to locate jobs in the schedule

Double Click Functionality:

- **Next Submit Dttm:** for timed jobs, a double click will display the execution window information for the job
- **Source Name:** a double click will display the source code (if available) from the current job
- **On Completion E-mail:** if populated, a double click will display the full field – useful if the display has been truncated
- **On Failure E-mail Override:** if populated, a double click will display the full field – useful if the display has been truncated
- **E-mail Content:** if not <NONE> then a double click will display the completion e-mail content

Right Click Popups:

- **Operator Control - Information:** by right clicking on the Information field when resolving a failed job a list of 'Autotext' options appears. This is a list of common reasons for job failures that can quickly be selected rather than re-typing each time

4) Job Parameters and Dependencies (blue): for the currently selected job, displays the parameters the job takes and any dependent jobs

Right Click Popup Menus:

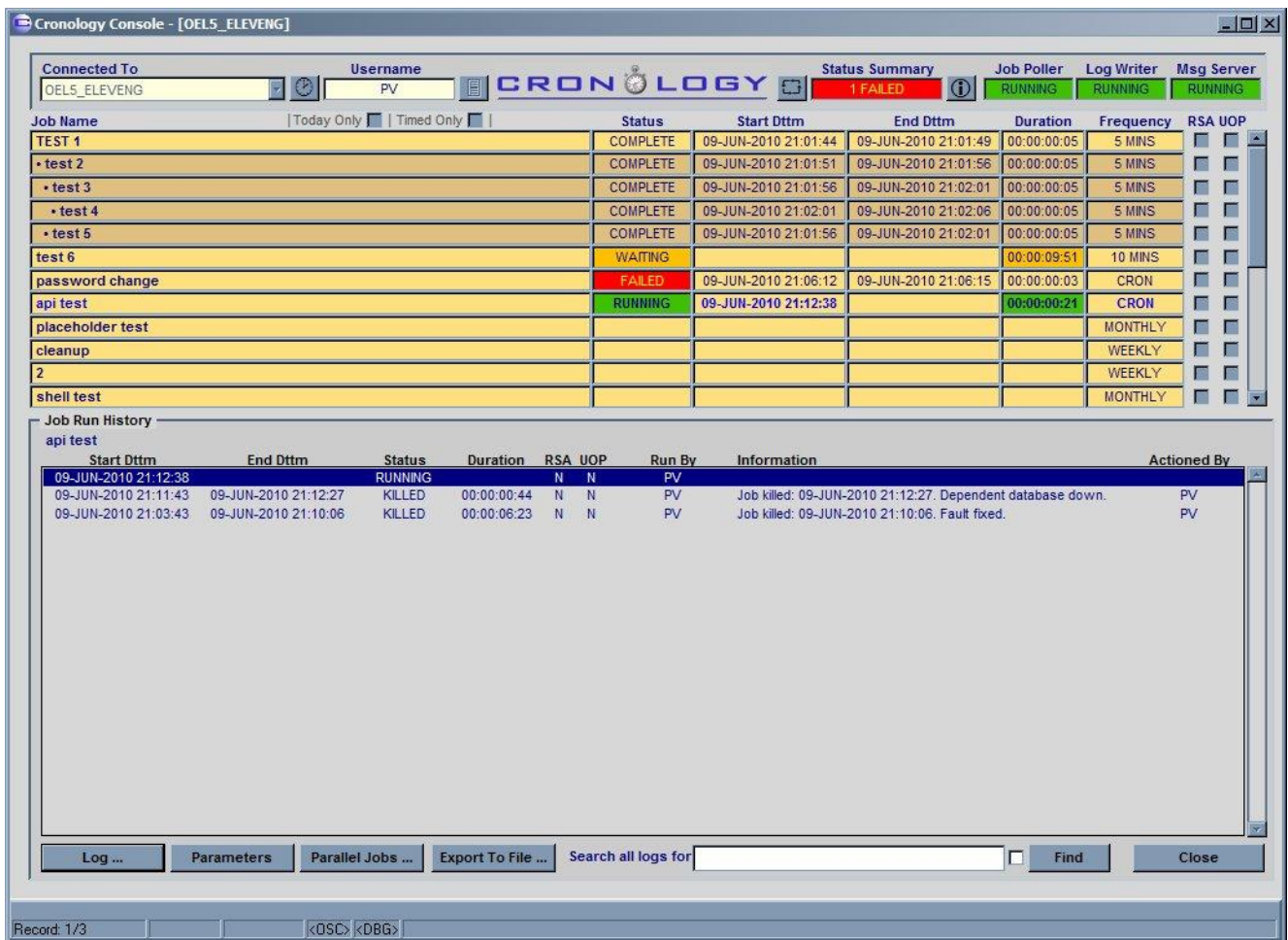
- **Name:** by right clicking on the parameter name field a popup menu of actions becomes available
- **Parameter Value:** a clipboard COPY option exists for this field

Double Click Functionality:

- **Job Name:** a double click will jump the display to the dependent job

4.3 Navigation and Default Actions

When viewing the main screen, the Run History button is the default action when the user presses the enter key. Therefore the user can navigate up and down the schedule using the arrow keys (or page up and page down), then press return to view the selected jobs run history:



Here the Log button is the default action when the user presses return. The user can therefore navigate up and down the run history and simply press return to view the job log for that run:

CRONOLOGY
 - VERSION 4.1 -
 - CONSOLE USER GUIDE -

The screenshot shows the Cronology Console interface. At the top, it displays 'Connected To: OELS_ELEVENG' and 'Username: PV'. The 'Status Summary' shows '1 FAILED' in red. Below this is a table of job schedules:

Job Name	Status	Start Dttm	End Dttm	Duration	Frequency	RSA UOP
TEST 1	COMPLETE	09-JUN-2010 21:01:44	09-JUN-2010 21:01:49	00:00:00:05	5 MINS	<input type="checkbox"/>
• test 2	COMPLETE	09-JUN-2010 21:01:51	09-JUN-2010 21:01:56	00:00:00:05	5 MINS	<input type="checkbox"/>
• test 3	COMPLETE	09-JUN-2010 21:01:56	09-JUN-2010 21:02:01	00:00:00:05	5 MINS	<input type="checkbox"/>
• test 4	COMPLETE	09-JUN-2010 21:02:01	09-JUN-2010 21:02:06	00:00:00:05	5 MINS	<input type="checkbox"/>
• test 5	COMPLETE	09-JUN-2010 21:01:56	09-JUN-2010 21:02:01	00:00:00:05	5 MINS	<input type="checkbox"/>
test 6	WAITING			00:00:10:26	10 MINS	<input type="checkbox"/>
password change	FAILED	09-JUN-2010 21:06:12	09-JUN-2010 21:06:15	00:00:00:03	CRON	<input type="checkbox"/>
api test	RUNNING	09-JUN-2010 21:12:38		00:00:00:56	CRON	<input type="checkbox"/>
placeholder test					MONTHLY	<input type="checkbox"/>
cleanup					WEEKLY	<input type="checkbox"/>
2					WEEKLY	<input type="checkbox"/>
shell test					MONTHLY	<input type="checkbox"/>

Below the table is the 'Job Log' for 'api test - 38 lines', which is currently 'RUNNING'. The log content is:

```

Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
Realtime output from PL/SQL via CRONOLOGY.API.LOG_LINE
  
```

At the bottom of the console, there is a search bar with 'Find', 'Find Next', and 'E-mail to' buttons, along with 'Mail Log' and 'Close' buttons. The status bar at the very bottom shows 'Record: 38/38' and '<DSC> <DBG>'.

Thus users can quickly view the latest log for a job by navigating to it in the schedule panel and hitting the return key twice. A double click on the job name in the schedule panel has the same effect.

4.4 Run History

As can be seen from the screen shot in the previous section, the run history panel has a number of buttons available:

- **Log:** show the job log for the selected run
- **Parameters:** display the parameters used for the selected run
- **Parallel Jobs:** display a list of jobs that were running in parallel with this run
- **Export to File:** create a CSV file detailing the run history of the job. Two modes are available, a full listing, or incidents only (i.e. runs that did **not** result in a status of COMPLETE)
- **Find:** search all job run logs for a particular search string (the checkbox is for matching case exactly – the tooltip on the checkbox states its function)

4.5 Job Log

As can be seen from the screen shot in the previous section, the job log panel has a number of buttons available:

- **Find:** search the log for a given string (the checkbox is for matching case exactly – the tooltip on the checkbox states its function)
- **Find Next:** find the next occurrence of the search string
- **Mail Log:** e-mail the job log to the specified recipients

4.6 Menu Structure

Main Menu Item	Sub Menu	Sub Menu	Sub Menu	Administ'or	Operator	Read Only
Schedule	Parameter Setup	Create		✓	x	x
		Amend		✓	x	x
		Delete		✓	x	x
		Delete Parameter History		✓	✓	✓
	Job Setup	Create		✓	x	x
		Bring Online		✓	✓	x
		Deleted Job History		✓	✓	✓
		Offline Job History		✓	✓	✓
	Change Control is ON OFF			✓	x	x
	Change History			✓	x	x
	Export Change History to File			✓	x	x
	Schedule Suspend is ON OFF			✓	✓	x
	Advance Schedule			✓	x	x
	Remove E-mail Recipients	All		✓	x	x
		Select		✓	x	x
	Export Schedule to File			✓	✓	✓
Database Sessions				✓	✓	✓
Status Summary				✓	✓	✓
Job Analysis	Activity Query			✓	✓	✓
	Activity Chart			✓	✓	✓
	Status by Occurrence	FAILED		✓	✓	✓
		KILLED		✓	✓	✓
		CANCELLED		✓	✓	✓
	Status by Percentage	FAILED		✓	✓	✓
		KILLED		✓	✓	✓
		CANCELLED		✓	✓	✓
Settings	Messaging Options			✓	✓	✓
	Console	Refresh Interval		✓	✓	✓
		Add Auto Text		✓	✓	x
		Remove Auto Text		✓	✓	x
		Clear Connection History		✓	✓	✓
		Use OPS\$ Logons is ON OFF		✓	✓	✓
		Users	Information	✓	✓	✓
			Access	✓	x	x
	Advanced	Settings		✓	✓	✓
		Change History		✓	✓	✓
Job Poller	Start Job Poller			✓	✓	x
	Stop Job Poller			✓	✓	x
	Job Poller Autostart is ON OFF			✓	✓	x
	Process History			✓	✓	✓
Log Writer	Stop Log Writer			✓	✓	x
	Troubleshooting	Diagnostic Test		✓	✓	x
		Show Lost Log Lines		✓	✓	✓
	Process History			✓	✓	✓
Message Server	Stop Message Server			✓	✓	x
	Message History	TODAY		✓	✓	✓
		LAST 7 DAYS		✓	✓	✓
		LAST 4 WEEKS		✓	✓	✓
		ALL		✓	✓	✓
		FAILED TO SEND		✓	✓	✓
		PENDING SEND		✓	✓	✓
	Process History			✓	✓	✓
Exit				✓	✓	✓

5 A QUICK EXAMPLE FOR ADMINISTRATORS

Lets create a SQL*Plus job that takes a single parameter and run it via the console. To run this example you will need to know the username and password for a test database account that has execute privilege on the DBMS_LOCK package.

1) Create the following SQL source file called cronology_test.sql in a directory of your choosing (remember the UNIX oracle user must have access to this directory):

```
prompt Anonymous PL/SQL block ...
begin
  cronology.api.log_line('Realtime output from PL/SQL block ... sleeping for 60 seconds ...');
  dbms_lock.sleep(60);
  cronology.api.log_line('The supplied parameter was: &1');
end;
/

prompt All done ... bye.
```

2) Logon to the Cronology Console and connect to your testing database

3) Create a parameter:

- Go Main Menu -> Schedule -> Parameter Setup -> Create
- Use the following attributes:
 - Name: Test parameter
 - Description: Test
 - Data Type: Character
 - Parameter Type: Fixed
 - Value: Hello World
- Click 'Save'

4) Create a job:

- Go Main Menu -> Schedule -> Job Setup -> Create
- Use the following attributes:
 - Job Name: Test job
 - Priority: 3
 - Job Description: Test
 - Source Type: SQL*Plus Script
 - Source Name: cronology_test.sql
 - Source Dir: <your directory>
 - DB Username: <test username>
 - DB Password: <test username password>
 - DB Connect String: <if the oracle account does not default to the SID of your test database then specify the TNS entry to points to your database here>
 - Schedule Type: TIMED
 - Next Submit Dttm: <double click the field for current data and time then add 1 to the year>
 - Job Frequency: ONE OFF
 - Schedule Type: TIMED
- Under the Job Parameters section click the 'Add ...' button
- Locate the parameter you created named 'Test parameter' and click OK
- Click 'Save'

5) Run the job:

- Ensure the job is selected by clicking on the job name on the main screen (the selected job will be bold blue)
- Click 'Execute ...'
- Click 'Execute'
-

6) Look at the job log:

- Click 'Run History ...' (or press return)
- Click 'Log ...' (or press return)

6 PARAMETER MAINTENANCE

As parameters are part of a jobs definition you typically create parameters before creating a job. The same parameters can of course be used by many jobs.

6.1 Creating a Parameter

To create a new parameter, from the main screen click on the Main Menu button (or press F6). Select Schedule -> Parameter Setup -> Create.

The Create Parameter panel will appear in the lower portion of the screen:

The screenshot shows the 'Create Parameter' dialog box. It includes a 'Name' field, a 'Description' text area, 'Data Type' and 'Parameter Type' dropdowns, a 'Copy Existing Parameter ...' button, a 'Value' text area, and an 'Override *' field. On the right, there is a 'Test Result' button, a help icon with text 'Fixed parameters and override values may include environment variables.', and a 'Conceal Values' checkbox. 'Save' and 'Close' buttons are at the bottom right. A '* Optional' label is in the top right corner.

Name: A unique name for the parameter

Copy Existing Parameter Button: click this button to select an existing parameter and copy its attributes into this new parameter definition, the name field will not be copied.

Description: A meaningful description of the parameter and its use

Data Type: Either 'Character', 'Date' or 'Number'

Parameter Type: Either 'Fixed' for literal values or 'SQL Statement' for dynamic / derived parameters. SQL statements provide a powerful means to obtain parameter values from anywhere in your database e.g. they could be values held in your own application tables, Oracle / application functions, Oracle sequences, sysdate / systimestamp etc.

Value: The value of the parameter. For Fixed parameters this will be a literal value. For SQL based parameters this field should contain the SQL statement that will be used to fetch the value. No trailing semi-colon should be specified. It is recommended you use the Test Result button to ensure your SQL statement behaves as expected.

Override: This optional field specifies the override value for the parameter. Override values can be used when executing jobs manually from the console (see later). The Job Poller process never uses override values. The override value may be changed either by amending the parameter or by changing its value when executing the job manually.

Conceal Values: Checking this box will mean the parameter value will appear as asterisks (*) on the main screen Job Run Parameters panel. This is useful if the parameter contains sensitive information that operators or read only users should not be allowed to see.

Test Result Button: pressing this button will evaluate the parameter and display its value. N.B. if your parameter is an Oracle sequence testing the parameter will cause the sequence to increase.

6.2 Amending a Parameter

Parameters can be changed by clicking on the Main Menu button (or press F6) and selecting Schedule -> Parameter Setup -> Amend. Select the parameter to be amended from the list of values, the parameter panel will then appear. This panel is the same as the Create Parameter panel except an additional button is present 'Show Parameter Uses'. This button is useful to quickly display all jobs that currently use the parameter.

6.3 Deleting a Parameter

Parameters can be deleted by clicking on the Main Menu button (or press F6) and selecting Schedule -> Parameter Setup -> Delete. A list of parameter will then be displayed. **Please note that only parameters that are not assigned to ANY job will be displayed in the list of values.** If the parameter you wish to delete is not displayed you must first unassign it from all jobs that are using it. Select the parameter and click 'OK'. Confirm the delete by clicking 'Yes' when the Delete Confirmation warning is displayed.

7 JOB MAINTENANCE

Important note: any jobs that have a source type of Shell, Perl, Java or Binary Executable are executed on the server as whichever UNIX user started the Oracle database. This is typically a user called oracle. As such it is vital that the oracle user has all privileges necessary to run the jobs source script e.g. if your job uses SFTP to copy files, then all the relevant SSH authentication will need to be setup for the oracle user. Likewise the oracle user must have read / execute privileges on the scripts you wish to run.

If the job you are about to create requires parameters, make sure you have created them / they exist before creating the job.

7.1 Creating a Job

To create a new job, from the main screen click on the Main Menu button (or press F6). Select Schedule -> Job Setup -> Create.

The Create Job panel will appear in the lower portion of the screen:

Job Name: A unique name for the job

Priority: A number to represent the priority of the job. This number is only used in failure message notifications.

Message Information: This button will bring up the following text entry box:

This field can be used to specify custom text in a message relating to the job (e.g. job failure). See the Message Templates section for more details.

Change History Button: click this button to view the change history for the job.

Documentation Link Button: click this button to specify a link to a document (e.g. web page on your intranet, document on your LAN) that relates to the job. This link can then be used by Console users to access any relevant information for the job

Copy Existing Job Button: click this button to copy the currently selected jobs attributes into this new job definition, the job name field will not be copied.

Job Description: A meaningful description of the job and its purpose

On Completion E-mail: An e-mail address to send a notification to on successful completion of the job. Separate multiple addresses with single space.

E-Mail Content: If a completion e-mail is to be sent, specify the content by selecting from:

- JOB LOG will simply e-mail the contents of the job log.
- MESSAGE allows you to specify a custom message, once MESSAGE is selected right click this field to edit the message

On Failure E-mail Override: specify an e-mail address to use instead of the default Operator E-mail address if the job fails. This is useful if you wish direct failure notifications for specific jobs to specific parties.

CC Operators: If you have specified an override address for failures but still want the default Operator E-mail address to be copied in then check this box.

Runtime Warning: Specify a runtime threshold time in days, hours, minutes and seconds after which a warning message will be sent. This is useful to notify operators if a job exceeds its expected runtime.

Source Type: Select a source type from one of the following:

- Binary Executable – any executable file that resides on the server. This could be an executable that is part of your application (e.g. compiled C code) or a UNIX command (e.g. rm)
- Java (Class / Jar)
- Perl Script
- Placeholder – as the name suggests this is purely a placeholder in the schedule. There is no job source associated with a placeholder, its acts as a dummy job. Placeholders are useful when you wish to group streams of jobs together e.g. you might put all your month end processing job streams as child jobs of a “Month End” placeholder. Any dependencies that month end processing has can then be set up on the placeholder job rather than repeating the dependencies for each stream of month end jobs.
- Shell Script
- SQL*Plus Script
- Stored Procedure

Source Name: the name of the source file / database procedure (this may need to be fully qualified depending on the user that connect to the database e.g. <OWNER>.<PACKAGE>.<PROCEDURE>

Source Directory: the directory location of the source file

DB Username, Password and Connect As: Specify a database username and password (case sensitive for Oracle 11g onwards) and optionally a SYS role:

- SYSASM
- SYSDBA
- SYSOPER

DB Connect String: Specify a valid TNS alias to connect to a remote database. If you leave this field blank Cronology will attempt a bequeath connection to the same database Cronology is running on. **If you are running a job on a remote database, Cronology must be installed on the remote database in order for the job to run correctly.**

Schedule Type: Select from either 'TIMED' or 'CHILD'. A timed job runs at a specific time (launched by the Job Poller process). A child job is launched only when its parent job completes successfully.

Resubmit After DB Stop Start: If a running job is interrupted by the database being shutdown this setting can be used to specify whether or not to automatically restart the job.

For TIMED jobs:

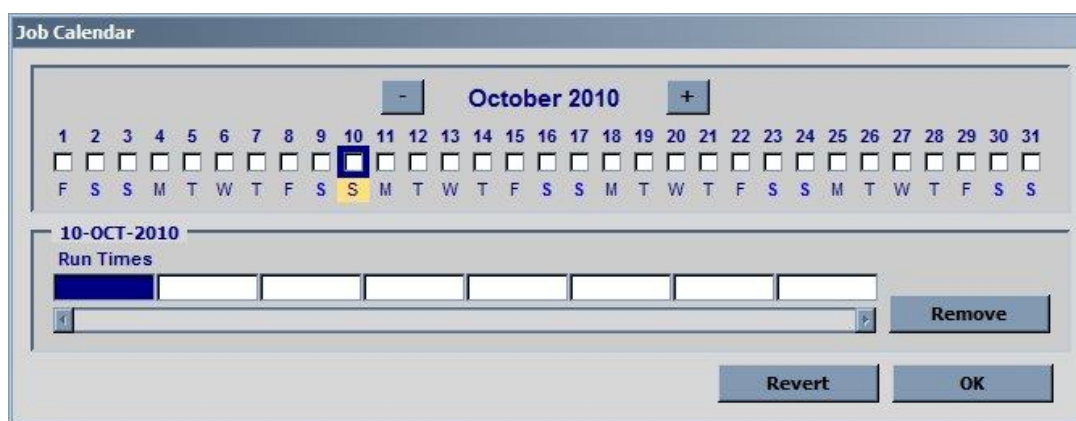
Next Submit Date Time: Specify the date and time you wish the job to next execute.

Window Start / End Date Time: If you only want the job to run for a specified period of time (e.g. you want it to run every 30 minutes but only for next week) then specify the window using the start and end date time fields

Job Frequency: Select a frequency for the job to run at. Frequencies are always based on the Next Submit Date Time, for example if you want to run a job every Monday (weekly) at 3:00pm, set the Next Submit Date Time to be 04-OCT-2010 15:00 (this date is a Monday) and set the frequency to 'WEEKLY'. Cronology has an extensive list of predefined frequencies, if none of these satisfy your requirements you can use a CRON, CALENDAR or SQL frequency to define when the job runs.

- 5 MINS – run the job every 5 minutes, starting from the Next Submit Date Time
- 10 MINS – run the job every 10 minutes, starting from the Next Submit Date Time
- 15 MINS – run the job every 15 minutes, starting from the Next Submit Date Time
- 20 MINS – run the job every 20 minutes, starting from the Next Submit Date Time
- 30 MINS – run the job every 30 minutes, starting from the Next Submit Date Time
- HOURLY – run the job every hour, the minutes past the hour at which to run will be determined by the Next Submit Date Time e.g. if you want the job to run **on** the hour make sure the Next Submit Date Time is on the hour e.g. 04-OCT-2010 15:**00**. To run the job hourly at a quarter past the hour the Next Submit Date Time would be 04-OCT-2010 15:**15**
- EO HOUR – run the job **Every Other** hour e.g. a Next Submit Date Time of 04-OCT-2010 15:15 would result in the job running at 04-OCT-2010 15:15, 04-OCT-2010 17:15, 04-OCT-2010 19:15 etc
- DAILY – run the job every day at the hour and minutes specified in Next Submit Date Time
- EO DAY – run the job every other day e.g. Monday, Wednesday, Friday, Sunday, Tuesday etc
- SUN-THU – run the job once a day from Sunday to Thursday only
- MON-FRI – run the job once a day from Monday to Friday only
- SAT-SUN – run the job once a day on Saturday and Sunday only
- WEEKLY – run the job weekly (once every 7 days)

- EO WEEK – run the job every other week
- MONTHLY – run the job on the same date each month. If the Next Submit Date Time is the last day of the month e.g. 30th September then the next run will always the last day of the month i.e. the job would run on 30th September, 31st October, 30th November.
- EO MONTH – run the job every other month
- QUARTERLY – run the job once every 3 months e.g. a Next Submit Date Time of 04-OCT-2010 20:30 would result in the job running 04-OCT-2010 20:30, 04-JAN-2011 20:30, 04-APR-2011 20:30 etc
- YEARLY – run the job once a year
- ONE OFF – a special case, run the job at the Next Submit Date Time **and never run the job again**
- CRON – specify a frequency in UNIX cron schedule format e.g. to run a job at 03:30 every day specify 30 3 * * *
- CALENDAR – select specific dates and times to run the job if there is no regular frequency to a job. The following window will appear to specify the run dates and time. **You must specify / maintain all the run times you require, Cronology does not (indeed cannot) automatically calculate the next run date time for Calendar based jobs**



- SQL – if you wish to schedule jobs based on data held in the database, specify a SQL query that returns a valid Oracle date type. Cronology will use this query to calculate the next run date time once the job has run

For CHILD jobs:

Parent Job: Select the job to act as a parent for this job i.e. this job will be launched only upon successful completion of the parent.

Job Parameters: Add any job parameters the job may require. Parameters do not have to be passed to the job, they can be present purely to maintain a job dependency (see later). If you do not wish to actually pass a parameter to the job but require it for dependency purposes uncheck the **Pass To Job** box. For Stored Procedure jobs a third button will be visible which will show the procedure specification i.e. you can see what parameters the procedure takes.

Job Status Overrides: There may be times when you wish to mark a job as completely or failing even though it has not done so. Once a job finishes (be it successfully or unsuccessfully) Cronology allows you to scan the job log generated by the job for keywords or phrases. You may force the job status to either FAILED or COMPLETE based on the search results. E.g. say the job fails with an ORA-01403 No data found error but you know that for this job this is OK to ignore you could set an override as follows:

Log Search String	Mark Job As
%ORA-01403%	COMPLETE

N.B. the use of standard Oracle search wildcards (% and _) can and should be used

If you add multiple overrides they are processed in the order they are listed. E.g. say the previous job also writes a sales revenue amount to the log, if the amount is zero you could force the job to fail by specifying:

Log Search String	Mark Job As
%ORA-01403%	COMPLETE
%Sales Revenue: £ 0.00%	FAILED

Even if the ORA-1403 is raised and detected, the zero value sales revenue is the last override to be processed and hence will supersede any previous overrides.

7.2 Amending a Job

To amend the currently selected job, on the main screen click the 'Amend ...' button in the Job Detail section (stack of buttons on the left hand side).

The Amend Job panel will appear in the lower portion of the screen. Fields are as described in Creating a Job.

7.3 Deleting a Job

To delete the currently selected job, on the main screen click on the 'Delete ...' button in the Job Detail section.

You will be asked to confirm the delete, please note that all historical job information (job run history and job logs) will also be deleted.

7.4 Killing a Job

When a job is RUNNING the job may be terminated immediately by clicking the 'Kill ...' button.

For Stored Procedure and SQL*Plus jobs, Cronology issues a kill session (immediate) command on the database session. For other types of jobs (e.g. shell scripts) the parent process and all spawned child process are killed using kill -9

7.5 Cancelling a Job

If a job is in a WAITING or SUSPENDED state (see later) the job may be cancelled by clicking the 'Cancel ...' button. This will prevent it from launching once its dependency is released.

7.6 Taking a Job Offline

If you wish to remove a job from the schedule but do not want to delete it, you can take the job offline by clicking on the 'Offline ...' button. All the jobs details will be retained but it will no longer appear in the Job Schedule section and will not be executed. Jobs can be brought back online via the Main Menu button (or press F6) -> Schedule -> Job Setup -> Bring Online.

7.7 Exporting a Job

By clicking the 'Export ...' button Cronology allows you to export the current job to another Cronology environment. Two methods are supported:

- **Export to Database** – this option allows a 'real-time' export of the job into another Cronology schedule. You will be prompted for connection details to the target database (Cronology must be installed and you must have valid Console access). The job will then be created on the target database.
- **Export to Release Script** – this option allows the job definition to be written to a SQL*Plus script. This script can then form part of your project release / code base. When using this option you will be asked if you wish to use any existing passwords for jobs or be prompted when the script is running (the script can always be edited to suit your needs later).

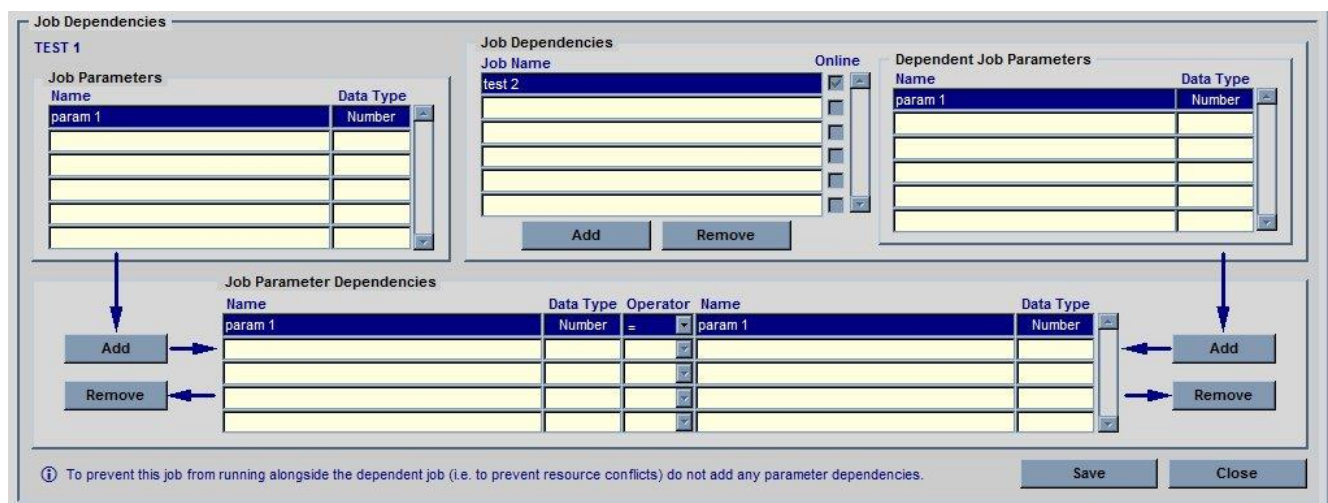
In both cases, if the job has child jobs you be asked if you wish to export these jobs also.

7.8 Managing Job Dependencies

Job dependencies are a fundamental part of the Cronology schedule. You control how and when dependent jobs are executed by linking them via their **job parameters**.

To manage the currently selected jobs dependencies click on the 'Dependencies ...' button in the job detail section.

The following panel will appear:



The screenshot shows the 'Job Dependencies' dialog box. It is divided into four main sections:

- Job Parameters:** A table with columns 'Name' and 'Data Type'. It contains one row: 'param 1' with 'Number' as the data type.
- Job Dependencies:** A table with columns 'Job Name' and 'Online'. It contains one row: 'test 2' with the 'Online' checkbox checked.
- Dependent Job Parameters:** A table with columns 'Name' and 'Data Type'. It contains one row: 'param 1' with 'Number' as the data type.
- Job Parameter Dependencies:** A table with columns 'Name', 'Data Type', 'Operator', and 'Name'. It contains one row: 'param 1', 'Number', '=', 'param 1', 'Number'.

There are 'Add' and 'Remove' buttons for each of the first three sections. At the bottom of the dialog, there are 'Save' and 'Close' buttons, and a note: "To prevent this job from running alongside the dependent job (i.e. to prevent resource conflicts) do not add any parameter dependencies."

The first section titled 'Job Parameters' simply lists the current jobs parameters. You may add parameters to the dependencies by selecting a parameter and clicking the 'Add' button directly below it.

To add a job to the dependencies, click the 'Add' button in the 'Job Dependencies' section. Select a job from the list and click OK. The parameter for the job you just selected in the 'Dependent Job Parameters' section. To add a parameter to the dependency click the 'Add' button below it. Once you have a pair of parameters for the dependency the only thing left to do is decide upon the operator. Parameters dependencies can either be =, <, >, >= or <=.

An example:

Say for example the system runs some month end processing. A parameter exists in the schedule 'Month End Date' – a number, taking values 201001 (Jan 2010), 201002 (Feb 2010), 201003 (March 2010) etc. Several jobs take this parameter, including job 'A' and job 'B'. To make job 'A' only run after job 'B' has completed you would simply edit job 'A's dependencies, add job 'B' as a dependent job, link the two 'Month End Date' parameters and add an = operator. By editing job 'A's dependencies you are making job 'A' dependent on job 'B' i.e. job 'B' must have completed successfully for the given month before job 'A' can run.

If you simply want to manage resource conflicts (i.e. job 'A' cannot run alongside job 'B') then add the job as a dependency but add no parameters. You will be prompted to create a mutual 'alongside' dependency on job 'B' back to job 'A' (recommended). In this case, whichever job executes first will prevent the other from running until it has completed.

8 EXECUTING JOBS MANUALLY

If you wish to run the currently selected job manually, click on the 'Execute ...' button in the job details section. The following panel will appear:

Execute Job

TEST 1

Description
A test job

Source Type
SQL*Plus Script

Source Name
test.sql

Source Directory
/home/oracle

Timing Information
Average Duration: 00:00:00:05
Runtime Warning:

Execute Options
 Use Override Parameters
 Run Stand Alone
 Later

Execute

Job Parameters

Name	Type	Data Type	Value	Override Value
param 1	SQL	Number		999

Close

Some options are available:

- Use Override Parameters (UOP) – if you wish to run the job with parameters values other than the standard values then check this box and specify any Override Values in the 'Job Parameters' section.
- Run Stand Alone (RSA) – if this check box is selected then no child jobs will be spawned upon completion of the job.
- Later – if the Job Poller is running you can specify a later time to execute the job, if you select this box the following will be displayed:

Execute Options

Use Override Parameters

Run Stand Alone

Later: 18-OCT-2010 12:09

Submit

N.B. Executing a job manually does not affect the scheduled run times of the job i.e. the Next Submit Date Time will NOT be affected by running a job manually.

9 SCHEDULE MAINTENANCE

9.1 Change Control

Cronology has a built in change control feature which can be enabled or disabled by an administrator. When enabled, if a user makes any changes to the schedule (e.g. create, amend jobs) they must also enter a change control entry for the change. Change history for entire schedule can be viewed from the Main Menu -> Schedule -> Change History. If you wish to view the change history for a specific job, select the job, click on 'Amend ...' in the job detail section and click on the Change History button:



9.2 Suspend the Schedule

If for any reason you need to suspend all job processing you can suspend the entire schedule from the Main Menu -> Schedule -> Schedule Suspend is ON/OFF. Suspending the schedule differs from stopping the Job Poller as stopping the poller only stops new TIMED jobs from being launched. Any child jobs of parents that succeed / any dependent jobs will still launch. Suspending the schedule prevents this.

N.B. suspending the schedule does not kill any currently running jobs – it merely stops all subsequent jobs from running.

9.3 Advance the Schedule

The entire schedule can be advanced to start at a given date and time in the future. For example, say you want to skip all this weeks processing and have the schedule start again next Monday. From the Main Menu, click on Schedule -> Advance Schedule. Enter the date and time you want the next job in the schedule to run. Cronology will then advance all jobs according to their frequencies such that they conform to the new schedule start date and time.

9.4 Remove E-mail Recipients

Over time jobs will be set up with custom e-mail recipients (e.g. On Completion / On Failure). When individuals move on or leave the company it would be very time consuming to examine each job and remove them from the recipient list. Cronology allows you to select individual e-mail addresses for removal from all jobs in the schedule. There is also the facility to remove ALL e-mail addresses from all jobs. This might be useful if the schedule has been imported from a live environment into a test environment and the live users should not receive e-mails from the test system.

9.5 Exporting the Schedule

Two options exist for exporting the entire job schedule:

- Documentation – choose this option to create a CSV file of the schedule. This file contains the following columns:
 - Job Name
 - Priority
 - Source Type
 - Source Directory
 - Source Name
 - Frequency
 - Next Run Date Time
 - Description
 - DB Username
 - DB Connect String
- Release Script – use this option if you wish to create a SQL*Plus script that will recreate the entire schedule. This is useful if you are cloning environments or want to include the creation of the schedule as part of a project release. When using this option you will be asked if you wish to use any existing passwords for jobs or be prompted when the script is running (the script can always be edited to suit your needs later).

10 MONITORING DATABASE SESSIONS

The Cronology console can be used to monitor sessions on the database including displaying explain plans for currently running SQL , tracing sessions (via TKProf) and viewing the TKProf output.

From the Main Menu select Database Sessions. The following screen will be displayed:

Database Session Information

Username	Logon Dttm	Status	TX Status	Inst.	OS User	Program Information
ADMIN	13-JAN-2011 12:58:25	ACTIVE	NONE	1	oracle	SQL*Plus sqlplus@localhost.localdomain (TNS V1-V3)
CRONOLOGY	13-JAN-2011 12:56:11	ACTIVE	NONE	1	oracle	Cronology Log Writer
PV	13-JAN-2011 12:56:26	ACTIVE	NONE	1	oracle	SQL*Plus sqlplus@localhost.localdomain (TNS V1-V3)
SCOTT	13-JAN-2011 12:54:59	INACTIVE	NONE	1	oracle	SQL*Plus sqlplus@localhost.localdomain (TNS V1-V3)

Highlight Cronology Related

Status **Sort (Mb)**

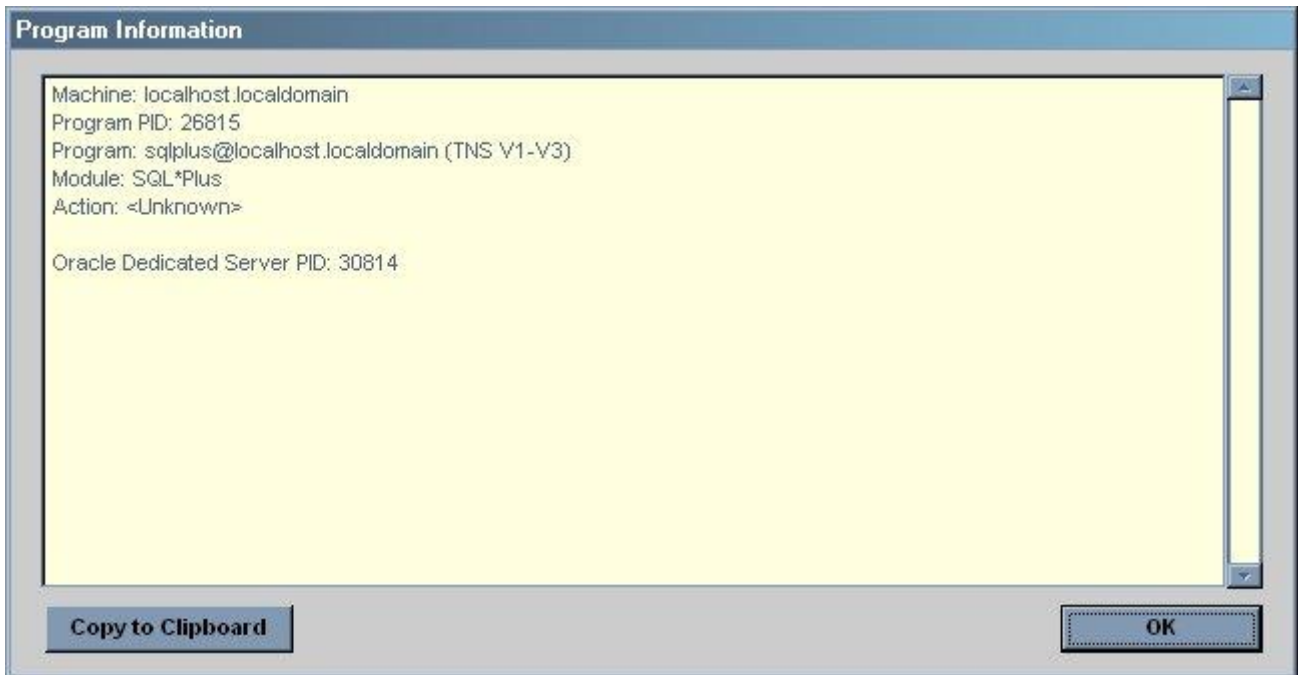

```

select count(*)
from dba_objects, dba_objects, dba_objects
  
```

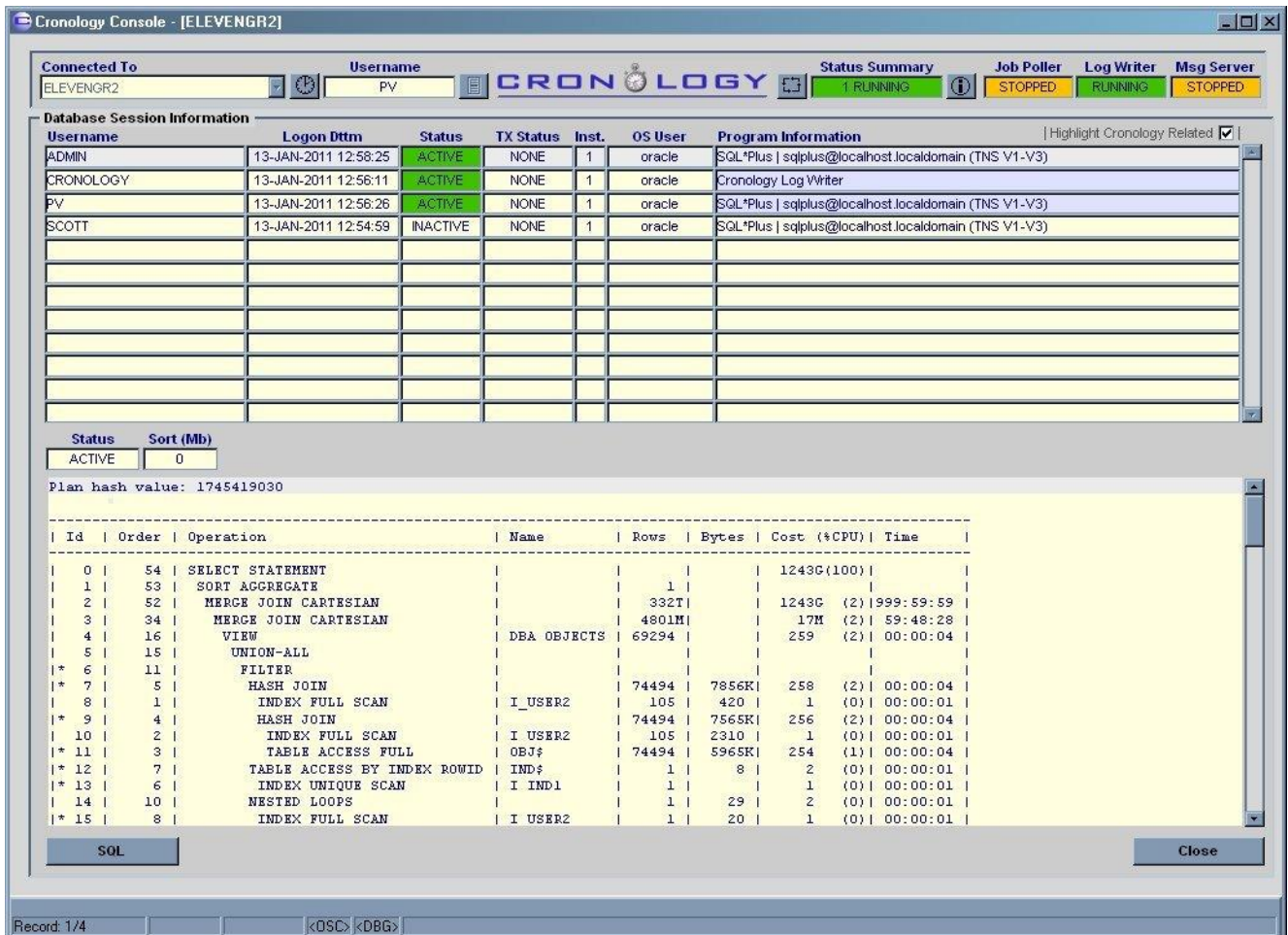

Record: 1/4 <OSC> <DBG>

The upper portion of the screen displays the sessions connected to the database. By checking the “Highlight Cronology Related” checkbox the Program Information field highlight any sessions that are either jobs launched by Cronology or Cronology related (e.g. server processes such as the Log Writer or Job Poller). This makes it easy to differentiate Cronology sessions from other user sessions.

By right clicking the Program Information field a menu appears with the option to display more information about the session, including operating system process IDs (PIDs), machine and module information e.g.



The lower portion of the main screen will display the active SQL for that session. By clicking the Explain Plan button this will show the current execution plan for the active SQL:



10.1 Username Menu

By right clicking the Username field the following menu appears:



N.B. for Read Only console users all menu options are greyed out.

Sessions may be killed (an immediate kill will be issued) or traced from this menu.

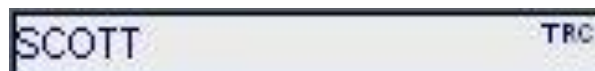
10.2 Kill Exclusions

If the session Username is in the following list then the Kill Session option will be greyed out:

'ANONYMOUS', 'CRONOLOGY', 'CTXSYS', 'DBSNMP', 'EXFSYS', 'LBACSYS', 'MDSYS', 'MGMT_VIEW', 'OLAPSYS', 'OWBSYS', 'ORDPLUGINS', 'ORDSYS', 'OUTLN', 'SI_INFORMTN_SCHEMA', 'SYS', 'SYSMAN', 'SYSTEM', 'TSMSYS', 'WK_TEST', 'WKSYS', 'WKPROXY', 'WMSYS', 'XDB', 'APEX_PUBLIC_USER', 'DIP', 'FLOWS_30000', 'FLOWS_FILES', 'MDDATA', 'ORACLE_OCM', 'PUBLIC', 'SPATIAL_CSW_ADMIN_USR', 'SPATIAL_WFS_ADMIN_USR', 'XS\$NULL', 'DMSYS', 'ORACLE', 'OPS\$ORACLE'

10.3 Tracing Sessions

If you select the Trace Session option an indicator will appear in the Username field to show that tracing has been enabled for that session.



Once tracing is enabled you can end the trace or view the trace output via the same menu:



Even when tracing is switched off the TRC indicator will still be shown for the session as the trace information can still be viewed via the Console until the session logs off. By clicking Show Trace Output, Cronology will show the TKProf output collected so far. N.B. if you turn tracing on **during** a long running SQL statement it may take some time for trace information to be written (i.e. Show Trace Output may display "Oracle cannot access the source file!").

CRONOLOGY
 - VERSION 4.1 -
 - CONSOLE USER GUIDE -

Cronology Console - [ELEVENGR2]

Job Poller: STOPPED
Log Writer: RUNNING
Msg Server: STOPPED

Connected To: ELEVENGR2
Username: PV
Status Summary: IDLE

Username	Logon Dttm	Status	TX Status	Inst.	OS User	Program Information
CRONOLOGY	13-JAN-2011 12:56:11	ACTIVE	NONE	1	oracle	Cronology Log Writer
SCOTT	13-JAN-2011 12:54:59	INACTIVE	NONE	1	oracle	SQL*Plus sqlplus@localhost.localdomain (TNS V1-V3)
SCOTT <small>TRC</small>	13-JAN-2011 13:03:40	INACTIVE	NONE	1	oracle	SQL*Plus sqlplus@localhost.localdomain (TNS V1-V3)

Trace Output

```

TKPROF: Release 11.2.0.1.0 - Development on Thu Jan 13 13:05:10 2011
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Trace file: /u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl_ora_26416.trc
Sort options: fchela exeela
*****
count      = number of times OCI procedure was executed
cpu        = cpu time in seconds executing
elapsed    = elapsed time in seconds executing
disk       = number of physical reads of buffers from disk
query      = number of buffers gotten for consistent read
current    = number of buffers gotten in current mode (usually for update)
rows       = number of rows processed by the fetch or execute call
*****

SQL ID: a5ks9fhw2v9s1
Plan Hash: 272002086
select *
from

```

Search for:
 Find
 Find Next

Record: 3/3
<OSC> <DBG>

1 1 JOB ANALYSIS

The Job Analysis menu (Main Menu -> Job Analysis) allows you to query the run history of the schedule.

11.1 Activity Query

Issue a query to see which jobs were running for a given time period.

11.2 Status By Occurrence

Display a list of FAILED, KILLED or CANCELLED jobs, ordered by number of occurrences. This can be used to quickly identify 'problem' jobs.

11.3 Status By Percentage

Display a list of FAILED, KILLED or CANCELLED jobs, ordered by a percentage of the times it has reached that status. E.g. a job that has only executed twice, and one of them failed, has a failure rate of 50%, a job that has executed 100 times and only failed once has a failure rate of 1%. This can be used to identify the reliability of jobs.

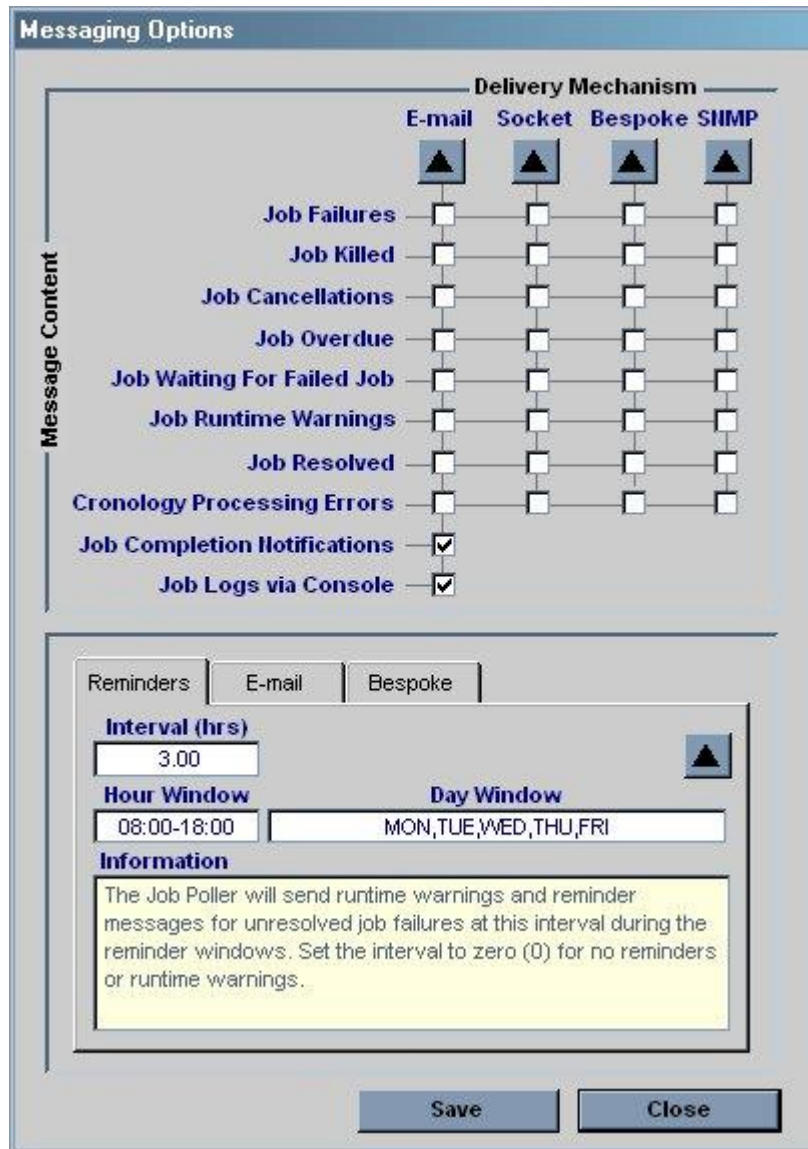
11.4 Average Durations

Display a list of all jobs and their average duration, ordered by duration. This can be used to spot the longest running jobs in your schedule.

1 2 SETTINGS

12.1 Messaging Options

The Messaging Options screen can be accessed from the main menu. The following panel will appear:



Cronology can send message via e-mail, sockets, SNMP or by calling a bespoke PL/SQL procedure within the database. You can control which events use which delivery mechanism by using the check boxes in the matrix.

The 'delta' buttons show the change history for each setting so you can easily track who has changed the settings over time.

The screen also allows you to set Reminder intervals (see above), set the operator e-mail address (used for all e-mail notifications) and also set the bespoke messaging procedure to use (if any):

Reminders | E-mail | Bespoke

Operator E-mail Address

<NONE>

Information

Operator e-mail address / distribution list. Please use the full address e.g. sysops@yourco.com. Separate multiple addresses with a single space.

Reminders | E-mail | Bespoke

Bespoke Messaging Procedure

<NONE>

Test

Information

Specify an Oracle stored procedure to be used for bespoke messaging. The procedure should take a single IN VARCHAR2 parameter which will be the message generated by Cronology. This message can then be used by the bespoke procedure as desired.

12.2 Console Settings

- **Refresh Interval:** set your console refresh rate (in seconds) – changing this setting does not affect other console users
- **Add Auto Text:** add text to the list of values that appears when an operator goes to update job information when resolving issues (list appears when right clicking on the Information field on the main screen).
- **Remove Auto Text:** remove text from the above list
- **Clear Connection History:** clear entries from the Connected To drop down list in the header section
- **Use OPS\$ Logons is ON|OFF:** Setting this option to ON will mean the console will automatically attempt to connect to the database using the OPS\$ prefix followed by your MS Windows username
- **Users**
 - **Information:** show information about others database user with Cronology Console access
 - **Access:** select database users to either grant Cronology Console access to, or edit / change their console access privileges

12.3 Advanced

- **Settings:** change advanced settings, these include the server temporary directory, the job log limit etc. Please liaise with your DBA when changing these settings
- **Change History:** display the change history of the advanced settings. All changes to advanced settings are captured (i.e. change capture cannot be turned off for advanced settings).

12.4 Message Templates

For added flexibility Cronology allows you to define templates for bespoke, socket and SNMP messages which override the default Cronology messages. Templates can be made up of fixed text and certain substitution variables, e.g. #JOB_NAME# in a template will be translated into the name of the job that triggered the message. Templates are configured via the CRONOLOGY.ADMIN database package (there are currently no screens in the console to maintain templates). Please see the Installation, Upgrade and Administration Guide for details of the SET_MESSAGE_TEMPLATE procedure within the CRONOLOGY.ADMIN package.

13 JOB POLLER

The Job Poller menu (Main Menu -> Job Poller) allows an administrator to stop and start the Job Poller. When Job Poller AUTOSTART is set to ON the Job Poller process will start up automatically when the database instance is started. The poller checks the schedule for jobs to run every minute.

A history of the Job Poller process (start-ups / shutdowns etc) is also available through this menu option.

14 LOG WRITER

The Log Writer menu (Main Menu -> Log Writer) allows an administrator to stop the Log Writer and perform a quick diagnostic test to determine the health of the Log Writer. Under normal operation the Log Writer will start up the first time a job is run, it will then remain running until either the database is shut down or an administrator stops it via the menu.

If problems are suspected, a diagnostic test can be run to 'ping' the Log Writer to check its health. Performing this action sends a message to Log Writer and waits for a response. A healthy Log Writer should return a response almost instantly.

If the Log Writer for any reason receives input for which it cannot identify a running job (i.e. it does not know which job log to write to) it will classify these lines of input as 'lost'. Lost log lines can be viewed via this menu. **On a healthy system there should be NO lost log lines.**

A history of the Log Writer process (start-ups / shutdowns etc) is also available through this menu option.

15 MESSAGE SERVER

The Message Server menu (Main Menu -> Message Server) allows an administrator to stop the Message Server and view message history. Under normal operation the Message Server will start up the first time a message is due to be sent, it will then remain running until either the database is shut down or an administrator stops it via the menu.

Message history can viewed via this menu, showing all message types, their status and when they were sent. This is useful to verify that certain messages (e.g. job completion messages) did actually get sent.

A history of the Message Server process (start-ups / shutdowns etc) is also available through this menu option.