Efficiency

Task Manager User Manual

Version 5.0

REVISED: MAR 2003

Table of Contents

Setting Up Tasks 5
Managing Tasks 5
Scheduling a Task 6
Checking the Schedule 8
Saving the schedule 9
Types of Tasks 10
Setting up an Executable task 10
Setting up a Function Task 10
Setting up a Report task 11
Setting up SQL tasks 11
Setting up Stored Procedure tasks 12
Setting up Triggers 12
Configuring Stored Procedures and Functions 14
Creating default values for Function and Stored Procedures 14
Running the Scheduler 16
Working with the Log 18
Viewing the log 18
Searching the Log 18
Resetting the log file 19
Deleting and backing up the task log files 19
Task Functions 21
Address DPID Update 21
Bulk Email 22
Yellow Stick Notes Maintenance 22
Update Account Rating 23
Update Contact Balances 23
Create Activities from emails 23
Email Users all emails that could not be made into activities 24
Update Membership Analysis 24
Send SMS 25
Global Import 25
Budget Functions 26
Update Budget History 26
Close Budget History 26
Create Budget History 27
Standard Procedures 29
Update Special Note Balances 29

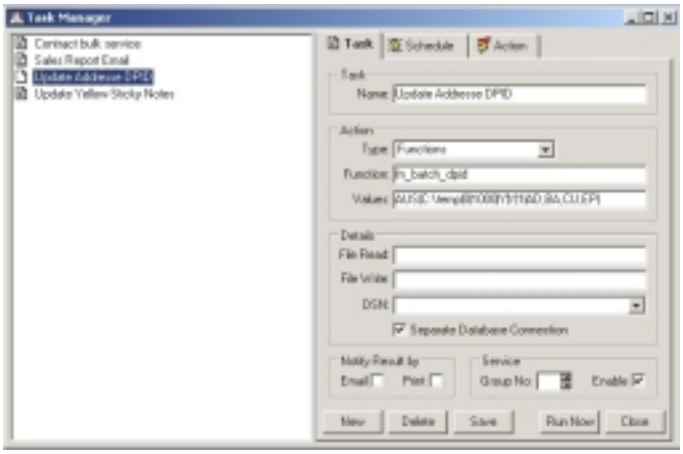
Create Bulk Service Log..... 29

Level 2, 11-15 Falcon Street, Crows Nest NSW Australia 2065
Tel: +61 2 9906 3505 • Fax: +61 2 9906 4424 • Eml: sales@avasystems.com.au
www.avasystems.com.au

Setting Up Tasks

Managing Tasks

Select Manage from the system Menu or click the Task Manager button, selecting the Task tab (Note: the 4 fields below the Type field change depending the type of task is run. See Types of Tasks for details).

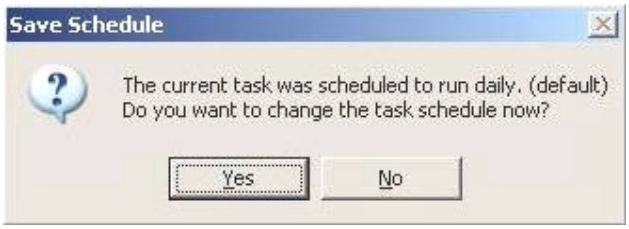


Creating New Tasks

Click New button and type the task name in the Name field.

Choose the task type from the Type drop-down menu. The Name and Attribute fields will then change depending on the task type. Reference to specific task types is later in this document on particular specific requirements.

Once all options are entered, save this task. This message box will pop-up:



By default, Task Manager schedules daily tasks when the task is saved. Click Yes to go to the scheduling tab to change the schedule.

Deleting a Task

Select the task you want to delete from the list of the left side of the window and press the Delete button.

Running a Task Now

Select the task required to run from the list on the left of the window and press the Run Now button.

Notifying Task result

Select Email and/or Print from the Notify Result By section

Field Name	Field Description
Email	This option will email a task result notification to recipients specified in Options. An email notification will be sent if Error Only is selected/
Print	This option will print a report from the default printer once the task is complete.

Running task on a different data-source

Select an ODBC from the DNS drop-down to select a data-source other than the one set-up in the application options.

Leave this field blank to run it on the data-source set-up in the application options.

Enabling Scheduling

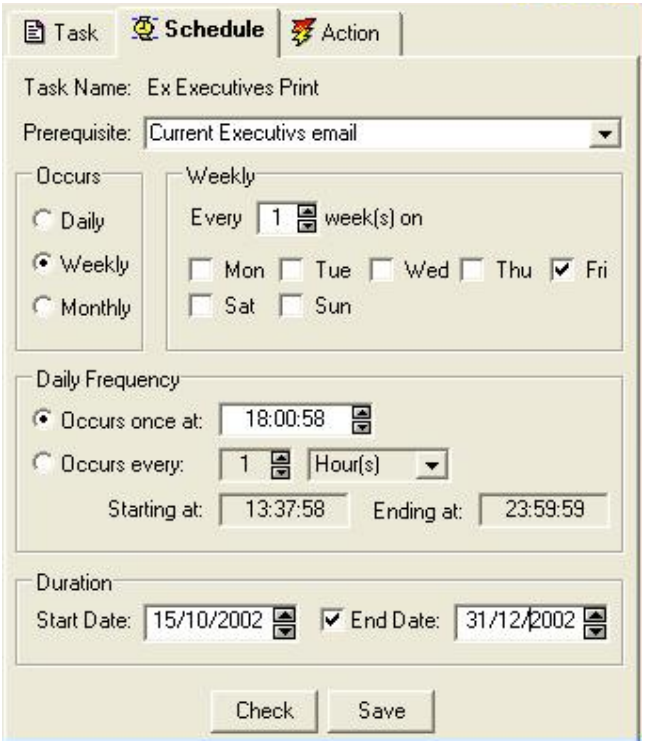
These Group No and the Enable options in the Service section need to be specified to schedule the task for another time.

Field Name	Field Description
Group No	All tasks can run by allocated group number when the scheduler starts. If unspecified, the group number will be 0 by default.
Enable	Check this box to enable this task to be scheduled. If this box is left unchecked it will not appear on the list of scheduled tasks when the scheduler is started.

Separate Connection

This option allows for multiple tasks to run using separate database connections without waiting for tasks to finish. Ensure you are licensed to run the number of desired tasks with a separate connection.

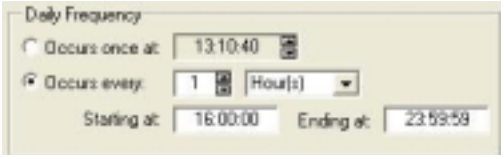
Scheduling a Task



A task can be scheduled once its been saved. It may then be set to occur once or regularly.

Important Note: The scheduler works on the 24-hour clock. If you wish to schedule a task to run a 5:00 pm you must set it to 17:00.

Field Name	Field Description
Prerequisite	Select this option when a task must run immediately before another. If there are no prerequisite tasks, leave this field blank.
Occurs	This option determines the frequency of the task - daily, weekly, or monthly. **Daily Options:**  Enter a number in the field to schedule the daily frequency of the task. The above example will schedule the task to run every 3 days. **Weekly Options:**
Daily Frequency	The daily frequency is used to determine how often the task will be run on a scheduled day. The two options here allow you to choose to run the task more than once daily.

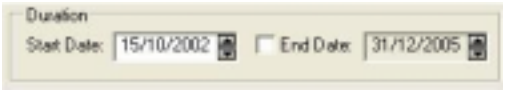


Occurs once at: Enter the time to start the task. If this option were chosen in the above example the task would run at 13:10:40 only. With this option chosen the Occurs every option will be disabled.

Occurs every: The scheduler can now run on the hour, minute or second. You must select from the drop-down menu run tasks every hour, minute or second, and enter a number in the field beside this for how often it will run. You then need to enter a start time in the Starting at field and the finish time in the Ending at field. The above example will run the task every hour from 16:00 to 23:59:59.

Duration

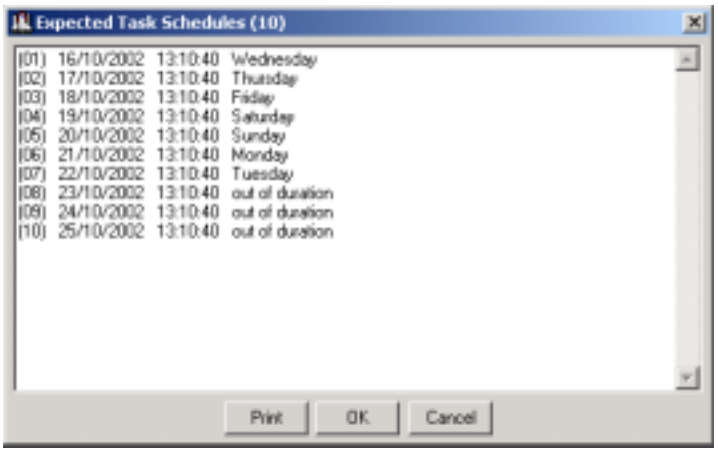
The duration frame allows you to decide what date to start the task and when, if ever, it will end.



Enter the required start date in the Start Date field. Check the end date box and enter a date in the End Date field to stop the task recurring at a specified date. Otherwise if you want the task to always be scheduled leave the End Date box unchecked.

Checking the Schedule

Once all scheduling options are entered, use the Check button to ensure the task is scheduled correctly. A pop-up window will display up to 10 tasks that will run when the scheduler starts.



The columns above relate to the following:

- **Instance:** 01 – 10, 01 being the first instance the task is run
- **Date:** The date the task will next run
- **Time:** The time the task will next run
- **Day of Week :** The weekday the task will run. If an end date was selected in the duration the dates after this end date will have out of duration for the day of the week.

Saving the schedule

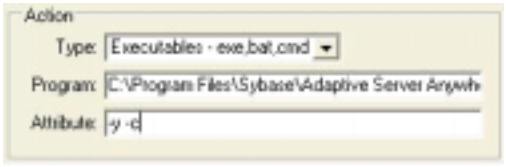
Click Save once you've verified the specified task times.

Types of Tasks

There are currently 9 task types that can be scheduled:

- Executables
- Functions
- Report – Email
- Report – Print
- Report – Save As
- SQL Enter:
- SQL File
- Stored Procedures
- Triggers.

Setting up an Executable task

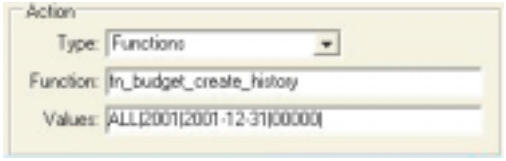


Go to Manage tasks and select Executables in the Type drop-down menu. Once this is chosen the Program and Attribute fields become available.

Double click on Program and browse to find it using the Open File dialogue box.

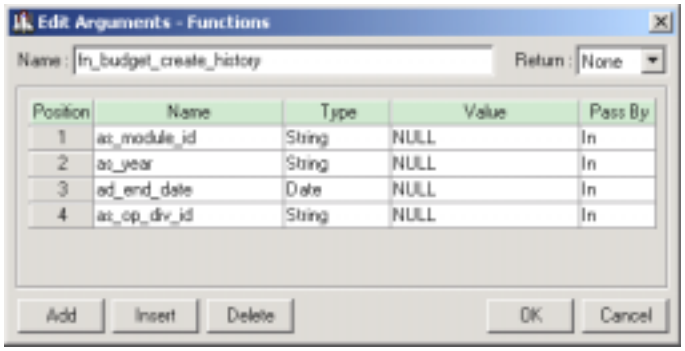
Enter any additional attributes the program needs to run correctly in the Attribute field.

Setting up a Function Task



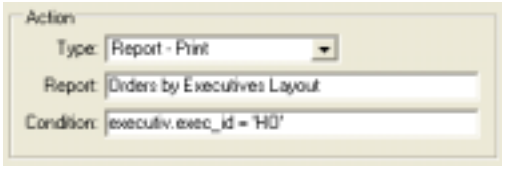
Select Function in the Type field. When this is chosen the Function and values fields become available.

Select the required function in the Function field.



If the function requires input parameters, double-click on the Values field and enter the appropriate values for the function in the values column.

Setting up a Report task



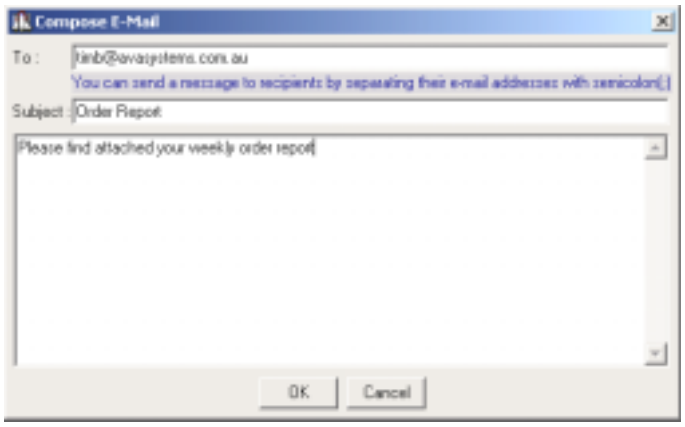
Select Report – Print, Report – Email or Report – Save As from the Type field. When these options are chosen, the Report and Condition fields become available. Select the reports you require.

Make the necessary selection criteria in the clause to refine your search condition. See above picture for example of this format.

The File Write field becomes available when Report – Save As is chosen, Double-click on File Write and choose the save location for this report with the Save File dialog box.

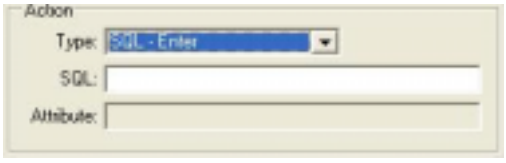
When Report – Email is selected, the Compose and Attach fields become available.

To send the report, enter the recipients email address in the Compose field. Double-click on the Attach and use Save File box to select a location and the report attached to the email will be saved.



Include an email message with the report email by double-clicking on Compose.

Setting up SQL tasks

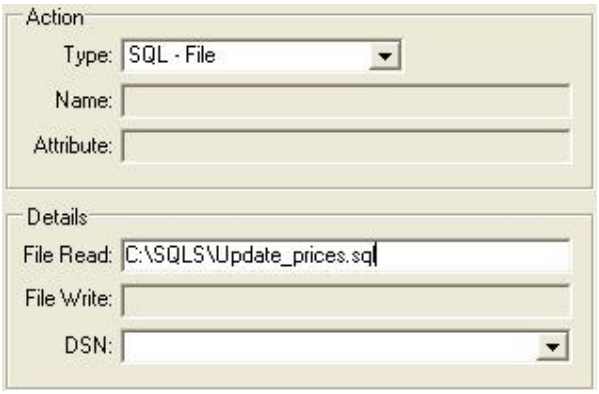


Select the SQL – Enter option from the Type field. When this option is chosen the

SQL field becomes available.

Double click on the SQL field to bring up the SQL Enter window and enter the SQL you wish to schedule.

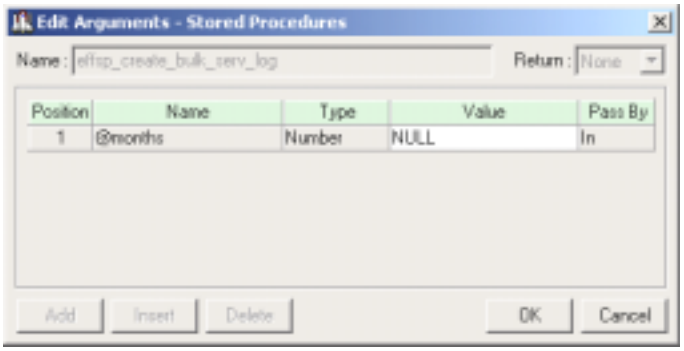
Note: The Task Manager will not show the results of SELECT statements. Only statements that will make changes to the database would be worthwhile scheduling.



When SQL – File is selected, the File read field becomes available. Double-click on the File Read field and find the SQL file containing the SQL required to schedule.

Setting up Stored Procedure tasks

The Procedure and Values fields become available when Stored Procedures is selected from the type menu.



Double-click on the Values field and enter the values in the values column if the stored procedure requires input parameters.

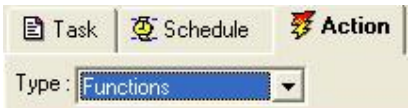
Setting up Triggers

Select Triggers from the Type field and Field No: will become available.

The field number refers to trigger fields (trigger_field_1, trigger_field_2..trigger_field_30) in the efftasks column. When the task is run it will insert the date and time in the specified trigger_field. You can then write a trigger for this trigger field column to fire every time this task is run.

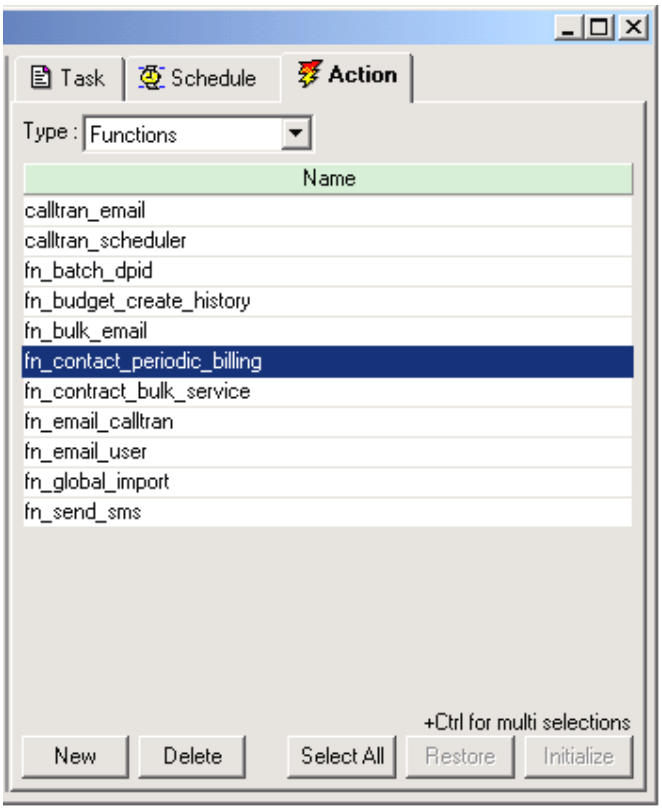
Configuring Stored Procedures and Functions

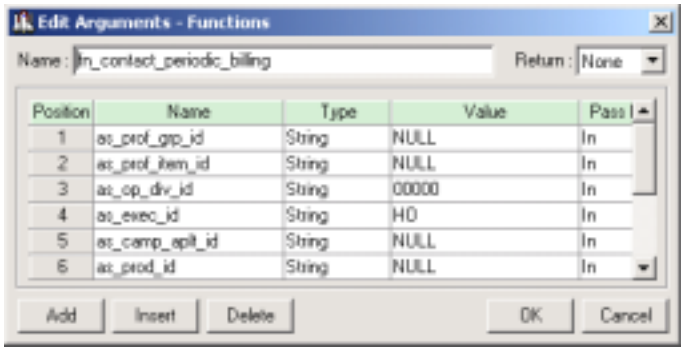
Click the Action tab in the Manage Task window to access the Functions and Stored procedure settings.



Choose to configure Functions or Stored Procedures in the Type drop-down menu.

Creating default values for Function and Stored Procedures





To configure the parameters for the function or procedure, double click on it. You will be presented with a parameter box where the values can be entered. Refer to the function or procedure for explanations of these values.

Creating New Functions

Select Functions in the Action tabs Type field and New to create the database functions definition. Development Kit is required to physically create a function.

Other buttons on Edit Arguments – Functions window do the following:

Button	Description
Add	Add a new argument after the last argument
Insert	Insert new argument at the position you last clicked
Delete	Delete the argument.

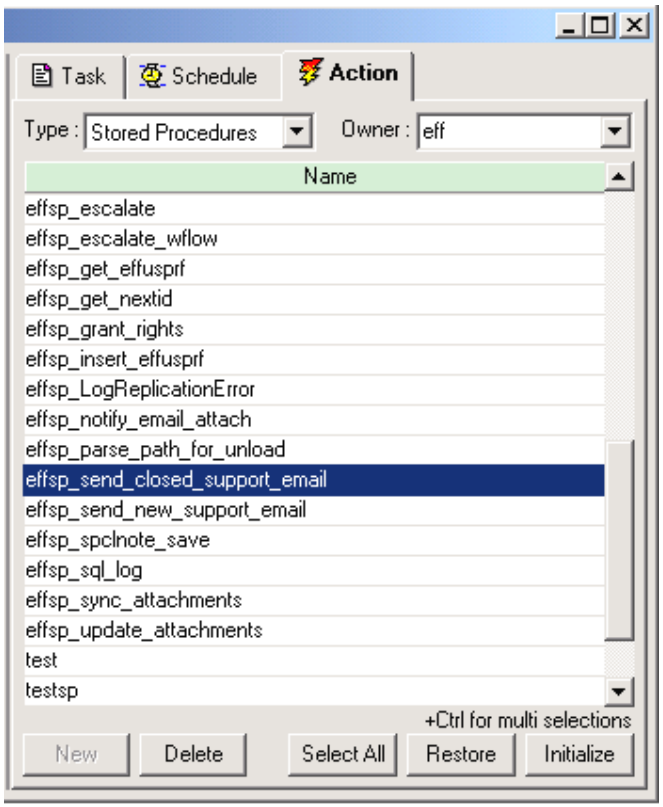
Ensure the function name, argument name, type, pass type and return value are identical to that in the program or the function won't work.

Deleting Functions

Select the particular function and click the delete button. To delete multiple functions, press Ctrl button while selecting.

Note: This will only delete the functions definition from the database. If its needed again add it again using the New button.

Stored Procedure editing

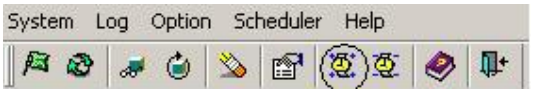


The Owner field, Restore, and Initialise buttons become available when Stored Procedure is selected from the Action tabs type field.

Owner refers to the owners of the stored procedures. The most often used stored procedures will be owned by eff.

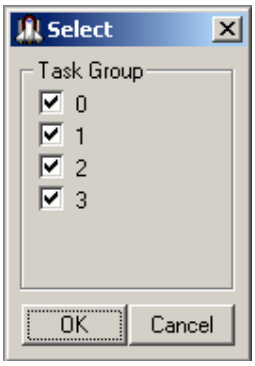
Button	Function
Initialise	This button will copy the procedure definitions into the effprocs table from the database. This should be used to initially synchronise the effprocs table to the database and when either adding or editing procedures to the database.
Restore	This button will update the definition of selected stored procedures and should be used if the stored procedures input or output arguments change, or it will run incorrectly.

Running the Scheduler



Select Start from the Scheduler menu or click the Start Scheduler button.

Select the task group(s) you wish to schedule and click OK. All groups unchecked won't start.



The scheduler will start in the background. To view the scheduled tasks and the scheduler activity, select Monitor from the System menu or click the task monitor button:

This window will display:

The current date and time is displayed at the top of the window. It will indicate whether the scheduler switched on and if a task is about to run. No tasks will start if the Scheduler is off.

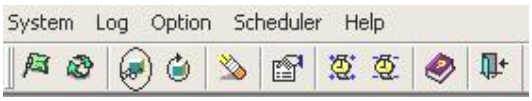
Column Name	**Function**
Group	The task group number.
Task	The task name entered in the Task Manager screen.
Next Run Time	The next time the task is scheduled to run in the format DD/MM/YY hh:mm:ss
Last Run Time	The last time this task ran since the scheduler was last started in the format DD/MM/YY hh:mm:ss. If the task hasn't run since the scheduler started the last run time will be set to 00/00/00 00:00:00
Status	When the scheduler starts, the task status starts as blank until it is time to be scheduled and then it's set to Open. This will tell the Runner the task needs to be run; then the status is set to Run. The status will be set to Close when finished, and the Log is viewable.

Result	The result starts as blank when scheduler starts. When a task is complete, it will have a Result of OK if the task ran successfully or Error if a problem occurred.

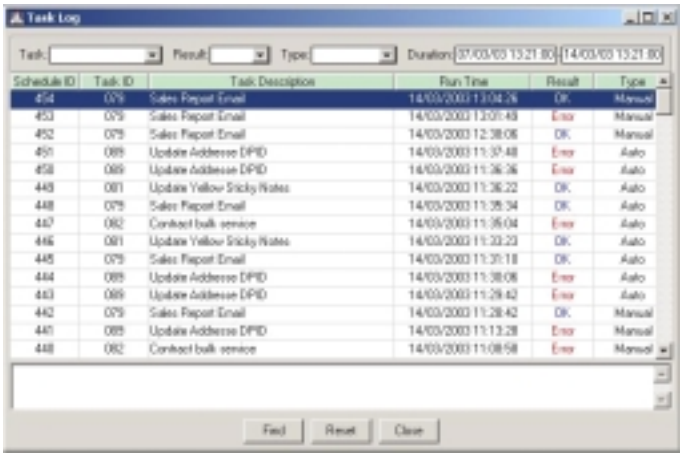
Working with the Log

Viewing the log

Select View from the Log menu or click the View Log button



This window will appear:



All log information is now stored in the effschlog table. The window shows all closed tasks in this table. The columns in showing on this window refer to the following:

Column	Description
Schedule ID	Unique Id identifying this scheduled item
Task ID	Unique Id identifying this task
Task Description	The name of the task entered in the task management screen
Run Time	The time the task ran
Result	The tasks result. If the task was successful it will show OK and if an error occurred it will show Error
Type	There are three possible Types: - Auto – Was scheduled and ran automatically - Manual – Was run manually in the task management screen by clicking Run Now - Prerequisite – Ran as a prerequisite task of another.

Searching the Log

Logs have these search fields, each can be used in combination of each other:

Field Name	Function
Task	Click to find a drop-down of the task names to sort by
Result	Either OK or Error, to find only the ones that either succeeded or failed
Type	Will either by Auto, Manual or Prerequisite

Duration	The data and time range to search. By default it will search within the last week.

Buttons

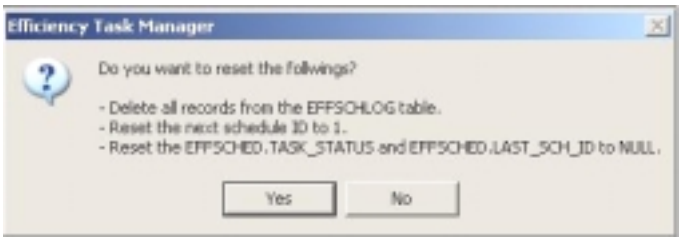
Button	**Description**
Find	Finds all logs based on the search criteria
Reset	Resets the search results and the search criteria, and resets the Duration fields to the last week
Close	Closes the window

Resetting the log file

To remove void logs, reset the log. This will enable quicker log searches.

Select Reset from the Log menu or click the Reset log button.

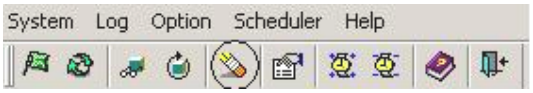
This message box will pop up:



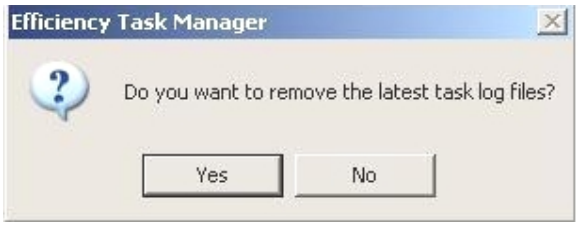
Clicking Yes will delete all the values in the logging table effschlog, set the next schedule id (unique id for the sffschlog table) to 1, and reset the status of all scheduled tasks to Null. This means any tasks with OPEN statuses scheduled to run, will be nullified and won't run. Any tasks currently running will also have their RUN status set to NULL and will not be logged. It is recommended this is only done when there are no Open or running tasks.

Deleting and backing up the task log files

If Save Task Log as Files was chosen in Debugging options, this option will allow you to delete and back up task log files no longer needed. To do this select Delete from the Log menu or click the Delete and Backup Log Files button:



This message box will pop up:



Choosing Yes will initiate this message box:

Click Yes to copy taskrunlog.txt and the taskschlog.txt files. They will be renamed by current date formatted as ddmmyy, (e.g. taskrunlog.txt for 17/03/03 is renamed the to taskrunlog-170303.txt.), and the files will be deleted. Click No to only delete the files.

Task Functions

Address DPID Update

Fn_batch_pdid

Reads the preset list of tables below and compares addresses against DRAT software (sold separately). If verified o.k, it fills in the DPID number, barcode, and sortplan. If DRAT software is unavailable, it's possible to set arguments appropriately to check addresses for formatting errors. This will leave the DPID, barcode and sortplan fields as they are.

Tables: address, bank, branch, compsite, contact, envengues, execprvt, media, orders, payment, rtrnprts, supptran, venubldg.

Note: When installing DRAT Version 2003.2.100, you must select Active-X EXE not Active-X DLL in Advanced Options. If you are installing DRAT version 2003.1.95 or earlier, you must select to use the out of process OCX (.exe).

Arguments:

as_countries	**String**	Comma delimited string of country ID's to check (Default AUS)
as_directory	**String**	Directory for storing log files. Error reports are recorded here for addresses failing verification (Default c:\)
al_records	**Number**	Maximum number of records to process for each table, 0 = all (Default 0)
al_commit	**Number**	Number of records to process before committing (Default 100)
as_use_drat	**String**	Y = use DRAT (must have DRAT installed) N = use Efficiency addres format check only (Default N)
as_use_addr_abbr	**String**	Y = use abbreviation. Systems will substitute address abbreviations on validated addresses. N = use description (Default Y)
as_log_level	**String**	1 = No table update, log message 2 = Update table, no log message 3 = update table, log message Messages in logs refer to warnings of possible invalid addresses found. (Default 1)
as_exc_codes	**String**	This field can be used to choose the tables you don't want to update. If this field is left blank, it will update all tables containing addresses. Excluded tables are represented by comma separated codes e.g. to exclude address, orders, and

payments, AD,OR,PA would be entered.

The codes for each table are:

Table	Code
Address	AD
Bank	BA
Branch	BR
Compsite	CU
Contact	CO
Evengues	EG
Execprvt (exec private)	EP
Media	ME
Orders	OR
Payment	PA
Rtrnprts(returne parts)	RP
Supptran	SP
Vendor	VE
Venubldg (Venue Building)	VB

Bulk Email

Fn_bulk_email

Reads through the calltran table and searched for records where:
The camp_splt_id matches the argument camp_id from the corresponding campsplt table, the ncall_date matches the current date the function is run and the call_stat is Open then generates an email to the recipient, being the call_cont_id with the relevant information contained in the remarks field of the particular camp_splt_id attached to this calltran record.

Arguments:

as_camp_id	**String**	Camp_id the calltran camp_splt_id needs to belong to.
as_ncall_type_id	**String**	Value of the ncall_type_id the records need to be.
As_next_action_mode	**String**	A= Append. Each subsequent run will append the remarks to the existing calltran so that a single calltran will contain each instance of an email being sent. C= Close. Each run will close the original calltran record and create a new calltran record for the next run sequence
As_end_action_mode	**String**	C= Close. After the last camp_splt_id is run, stops the campaign and closes the activity R= Restart the sequence starting from the first camp_splt_id
an_interval	**String**	>0= The number of days the ncall_date will be set forward for the next run 0= Sets the ncall_date relative to the original call_date and the date the function ran and makes the ncall_date the same interval.

Yellow Stick Notes

fn_update_spclnote_expired

This function scans the spclnote table and deletes notes where the valid_to date is

Maintenance

less than or equal to today.

Arguments: None – returns number

Update Account Rating

Fn_update_account_rating

Updates the account ratings (accnt_stat, stat_rate_id) for the company and contact tables using the user defined matrix in the accnudph and accnupd tables.

Note: Some combinations are invalid, such as choosing CO for contacts and O for Orders. Orders are unavailable for contact management.

Arguments:

as_type	**String**	CO = Updates the contacts table CU = Updates the compdivs table ALL = Updates both of the above
As_mode	**String**	O = checks values in orders table S = Checks values in salehist table
As_tag	**String**	Audit1 = Create fieldaud row if accnt_stat_id changed Audit2 = Create fieldaud row if stat_rate_id changed Audit12 = Both of the above

Update Contact Balances

fn_update_contblns_overdue

This function should be run on the last day of each month. It updates the current and overdue balances in the contblns table. It clears the current balance (curr_bal) and slides the balances down one month. This means the value previously in curr_bal will now be in over_due_1, the value previously in over_due_1 will be in over_due_2, the value previously in over_due_2 will be in over_due_3 and the value previously in over_due_3 will be added to the current value in over_due_4. If the balance is negative it adjusts the balance from bottom up i.e. reducing the oldest balance first.

Arguments: None – returns number

Create Activities from emails

fn_email_calltran

Creates activities in the calltran table from the emails in the Eff folder in Outlook that was specified in the Email/Activities tab in Options. The email will be deleted when the activity is created. To run this function, a public folder must be set-up in Outlook and set to copy email with certain criteria, i.e. subject keyword etc, to this folder. Efficiency email account set-up in Task Manager's role must be set to this folder's Owner so it can read, write, and delete every email it contains. This error will return if the permissions are set-up incorrectly:

You do not have sufficient permission to perform this operation on this object. See the folder contact or your system administrator. [Microsoft Exchange Server Information Store - [MAPI_W_PARTIAL_COMPLETION(40680)]].

We recommend all other users of this folder have an Author role, which will only allow them to create, edit and delete emails addressed to them.

Note: This function requires Microsoft Exchange server be selected in the

Emailing Options.

If the email is to a person whose email address is not set up as an Executive in Efficiency an activity cannot be created and the email will be deleted from the Eff folder.

If the email is to multiple recipients, it will use the first address in the To field. Ensure an executive is set up for each email address people may send to and want to create activities for. If no executive is found with this email address the email will be deleted. The contact for the activity is determined from the From address. If no contact was found with that email address, the subject will be appended with the word [PROCESSED]. For example, an email with the subject "Efficiency training" would be appended to "[PROCESSED]Efficiency training". To avoid this, ensure all contact email addresses are updated in Efficiency if they are changed.

Arguments: None – returns string

Email Users all emails that could not be made into activities

fn_email_user

Notifies users via email of emails that couldn't be made into activities from the fn_email_calltran function. It searches the Eff folder set up in Email/Activities tab in Option for emails with [PROCESSED] in the subject. These emails are deleted, and a list of addresses and subjects of emails that couldn't be made into activities emailed. This information indicates which contact emails were not found in the database and then add or append this contact in Efficiency. The activity can be created email manually or the email recopied to the folder.

Note: This function requires Microsoft Exchange server to be selected in the Emailing Options.

It's recommended that when scheduling a task to use this function, that a task which uses fn_email_calltran runs prior to it. The task using fn_email_calltran should be the prerequisite task so it won't run if fn_email_calltran fails. Also, this function deletes unprocessed emails from the Eff folder. Users should be advised not to delete emails with activities created from in their inbox until this function is performed lest manual entry be required.

Arguments: None – returns string

Update Membership Analysis

Fn_update_membership_analysis

Creates or updates membership analysis data in the Memanlys table from the data in the member table. It creates rows for the country, state, year and month each grouped by combinations of memb_stat_id, memb_card_id, memb_club_id, memb_cat_id. For these groups the rows will show how many have reserved seats, voting rights, had cards printed, joined that month, had their memberships approved that month, became full members, died, were members since that month and have car park passes.

Arguments:

As_country_id	**String**	The Country id e.g. AUS
As_stat_id	**String**	States to look at separated by commas e.g. NSW,VIC
As_periods	**String**	The to and from date range of the analysis formatted as yyyymm-yyyymm e.g. to create or update analysis data from January 2002 to June 2002 enter 200201-200206

Send SMS

fn_send_sms

This function reads the smslog table and sends any unsent or previously busy SMS messages. SMS statuses are:

- **B = Busy** or **H = Hold:** status of unsent messages.
- **E = Error:** if an error occurs and the SMS is unsent.
- **T = Timeout:** if the function times out. Time-out is base on the Timeout field of the Comms tab in Options. It's recommended you set timeout to at least 90 seconds. Ensure the Timeout field holds a value or if there is a problem, it will never timeout.
- **S = Sent:** the message has been sent.

Note: If you are not using email to send SMS and are using a dial-up modem then this function must be run on the computer which has the modem connected to it

Arguments: None

Global Import

Fn_global_import

This function imports data from a file based on the function id set up in the Efficiency's Global Import utility. This function is useful for importing data from a specific location and file on a regular basis.

Note: You must certain the function id is set up correctly in Efficiency and can import the file from there to import it again with different data.

Arguments:

As_function_id	**String**	The function id set-up in the Global Import utility the file to import and which columns correspond to which columns in the database in the file. E.g. PLANT_STD have been set-up

Budget Functions

All transactions related relevant to budgets are stored in the temporary budget history table (temp_hist) until a budget update is performed.

These functions take the data from the temporary budget history table (temp_hist), and transfer it to the appropriate history type table as long as there is a budget already set up for it. The history tables are as follows:

Primary table	History Table
Campaign	Camp_hist
Company	Comp_hist
Executiv	Exec_hist
Product	Prodhist
Prodcat	Prdchist

The data in the following columns in the history tables should be updated with adding the values it obtains from the temp_hist table:

today_qty,today_value,wtd_qty,wtd_value,mtd_qty

mtd_value,ytd_qty,ytd_value

These represent the quantities and the $ values for today, the Week to date, the month to date and the year to date. The previous months and years quantities and values are stored in these columns:

month_1_qty, month_1_value, month_2_qty, month_2_value, month_3_qty, month_3_value, month_4_qty, month_4_value, month_5_qty, month_5_value, month_6_qty, month_6_value, month_7_qty, month_7_value, month_8_qty, month_8_value, month_9_qty, month_9_value, month_10_qty, month_10_value, month_11_qty, month_11_value, month_12_qty, month_12_value, year_1_qty, year_1_value, year_2_qty, year_2_value, year_3_qty, year_3_value

The following must be set up to perform these functions:

1. The Budget definitions for the campaign, company, executive, product or product category.
2. In the Shared/Supervisor/Global Options, set the Current Budget Year
3. In the Shared/Supervisor/Global Options, set the Current Budget Year Start Date
4. In the Shared/Supervisor/Global Options, set the Current Budget Year End Date

These functions will only put data in the history tables is a budget definition is set up for that item and the transaction date is for the current budget year and between the current budget years start and end dates.

Update Budget History

Fn_update_budget_history

This function updates the history of all modules in the history tables with the data from the temp_hist table. This function should be run daily, weekly or monthly depending on how often you wish to check the Actual Vs Budget reports in Efficiency.

Arguments: None

Close Budget History

Fn_close_budget_history

This function closes the all the budget history records which means all budget history records action dates will be set to the last day of the budget year. The

action date is the act_date column in the budget history table and represents the last time the history line was updated. This function should be used if the last day of the current budget year has passed or if there will be no-one available to update the history when the last day of the current budget year arrives.

Arguments:

As_module_id	**String**	The module ID of the module you wish to close the history of. The options are: ▪ **ALL** Closes the history of all the modules ▪ **EX** Executive History ▪ **CP** Company History ▪ **CO** Contact History ▪ **PR** Product History ▪ **PRC** Product Category History
As_year	**String**	The history year you wish to close in YYYY format.
Ad_end_date	**Date**	The end date of the budget year which will be updated. This is what the action date of each record will be set to
As_op_div_id	**String**	The Operating Division Id.

Create Budget History

Fn_create_budget_history

This function resets the old budget history records to the last day of the current budget year and creates a new set of history records for the first day of the new budget year for each record in the old budget year. The current today, wtd, mtd and ytd values will be set to zero and the previous dates data will be updated with last years values.

Arguments:

As_module_id	**String**	The module ID of the module you wish to close the history of. The options for this are: ▪ **ALL** Closes the history of all the modules
As_old_year	**String**	The old budget year which will be closed.
As_new_year	**String**	The new budget year
Ad_start_date	**Date**	The start date. This is what the action date of each record will be set to. For the records of the old budget year it will be the end date, and for the new budget year records it will be the start date.

Ad_start_date	**Date**	The start date. This is what the action date of each record will be set to. For the records of the old budget year it will be the end date, and for the new budget year records it will be the start date.
As_op_div_id	**String**	The Operating Division Id.

Standard Procedures

Update Special Note Balances

effsp_maintain_spclnote_balance

This function updates contacts and companies yellow sticky notes which current and/or overdue balances with their balance in the special note. If their balance is zero if will delete the note. The data used is in the Contblns and Compblns tables.

Arguments:

keyword	**String**	This refers to what you would have typed in the description field of the Yellow Sticky Note. **Note:** If there is no data in the Contblns or Compblns tables for a sticky note with this keyword in its description, the note will be deleted. Make sure no one uses this keyword for notes not relating to the balance.
mode	**String**	Current = Only displays the Current balance in the notes Overdue = Only displays the overdue balances in the notes All = Will display both the Current and Overdue amounts in the notes.

Create Bulk Service Log

effsp_create_bulk_serv_log

This procedure creates support log and transactions in the supplog and supptran tables for contracts that are serviced regularly throughout the month and can then be used by you to find out the date which those contracts are due.

Note: **Before running this procedure it is recommended that you check the following columns for NULL or invalid data. All contracts with invalid data will be skipped.**

Bulk_serv_months

This column contains the months in which the contract is serviced. It contains numbers from 1 – 12 separated by commas. For example, a contract is to be serviced every month, the data will look like this:

1,2,3,4,5,6,7,8,9,10,11,12

Data such as the Month name, or numbers separated by a dot rather than a comma will be ignored.

Bulk_serv_logged_months

This column contains the months already logged this year. It too contains numbers from 1 – 12 separated by commas. For example, if a contract has been logged every month until May, the data looks like this:

1,2,3,4,5

Note: Because months logged this year are ignored, it will not log any contacts due next year. You must ensure the month parameter will only

include remaining months in the current year.

Bulk_serv_day

This column contains day of the week the contract is serviced. It contains numbers from 1 – 7, representing the days of the week, Monday represents 1, Tuesday represents 2, Wednesday represents 3, Thursday represents 4, Friday represents 5, Saturday represents 6, and Sunday represents 7. Any other data such as NULL or the word “MONDAY” will be ignored.

Bulk_serv_day_no

This column represents which week in the month the contract must be serviced. It contains the numbers 1 – 5 and –1. 1 represents the first week in the month, 2 represents the second, 3 represents the third, 4 represents the 4 week in the month, 5 represents the fifth week in the month and –1 means the contract will be serviced on every week in the month. We do not recommend you use 5 because there may for example not be a fifth Monday in the selected Month and it would therefore be ignored. If this data is any other value number or word, the data will be ignored.

Arguments:	**months**	**integer**	An integer from 0 – 12 representing how which month you wish to log. 0 represents the current month, 1 represents next month and so on. It will not accept negative numbers.