

# JOB Administrator Reference

Version 4.0 Level U

Copyright 2003 - 2006

Rosebud Management Systems

## JOBS system overview

Eden Server provides a complete batch system via several of its server side plugins, namely the Background Manager and the Scheduler. While a complete set of API's are provided to access all functions of the Eden batch system, Eden also ships with a completely integrated batch management system called 'JOBS'.

JOBS has been designed to work with Eden Server in a seamless fashion, taking advantage of the security configuration options in EManager as well as the functionality provided by the servers Background Manager and the Scheduler.

Functionally, JOBS is a series of user interfaces that are accessed directly from the Eden Client via toolbar buttons. Use of and access to each of the up to 4 client toolbar buttons is controlled based upon a users security settings which are defined in EManager.

A toolbar button is supplied for each of the following core functions of the JOBS system:

- Access to job submission features
- Access to the system Job Queue
- Access to the system Print Queue(s)
- Access to JOBS administrator functions.

Information and documentation on the first three of these items, job submission, the Job Queue and the Print Queue are detailed in the JOBS user reference guide. Information on set up and use of the overall JOBS system is detailed in this, the JOBS Administrators Reference guide.

Prior to using JOBS, it is suggested that a thorough understanding of the items covered in this document as well as the security setup sections of the EManager Reference guide be gained.

The JOBS system interacts with Eden's existing batch facilities, namely the System Job Queue/Job Scheduler and the System Print Queue.


The features and capabilities provided by JOBS, version 3.13, include:

- On the fly access to any JOBS feature by simply using the displayed Eden Client JOBS toolbar access buttons. The JOBS toolbar buttons may be used at anytime, even during the execution of another CICS transaction, upon completion of JOBS, the original user screen is restored.
- Full user-security driven access to system entities, including jobs, reports and the system queues.
- Ability to submit pre-defined batch jobs to Eden Server for processing. Submitted jobs may include user input (i.e., control card or command line parameters), scheduling information, pre-run and post-run dependencies, scheduling and run time

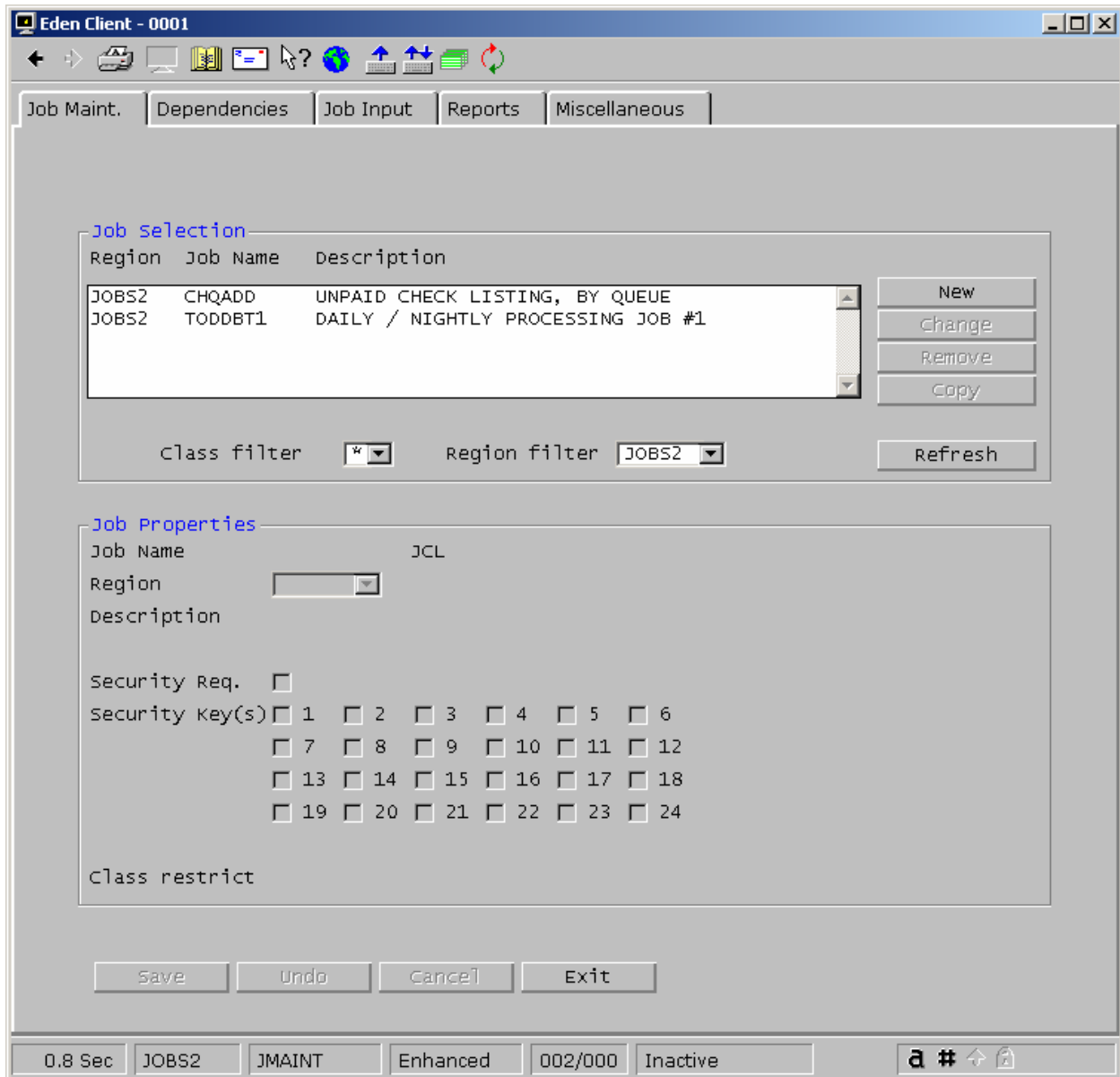
exception settings, event driven email notifications as well as other advanced features.

- Ability to view and or control Job Queue entries, based upon user security settings. The JOBS Job Queue interface includes a number of display filters allowing the user to display only those jobs of interest. Display filters are provided for job class, job name, job owner user name, job status as well as from and to dates. Depending upon the users security change access to individual Job Queue entries may also be available, allowing any parameter associated with the job to be changed.
- Ability to view, print or control Print Queue entries, based upon user security settings. The JOBS Print Queue interface also includes several display filters including: Print Queue job-id, From and To dates, base Job name, Eden Region name, Queue Entry class values, as well as Print Queue entry type, i.e., Public or Private.
- JOBS Administrators also have access to the JOBS administrator functions, which are detailed in this document.

## JOB Administrator Functions

The Administrator function of the JOBS system, deals with the definition of individual jobs to the JOBS system. It is important to realize that all jobs to be run via the JOBS system must first be defined by using the JOBS administrator displays. When a signed on user is determined to have Administrator authority, the Eden Client toolbar will include the button, which appears as: 

When the administrator functions are first started, the Eden Client window will switch to GUI display mode, if it isn't already in GUI mode, and will show the following start up screen. Note that the toolbar button for switching between GUI and Character display modes will remain disabled while using the administrator functions.



The screenshot shows the 'Eden Client - 0001' window with a toolbar at the top containing navigation and system icons. Below the toolbar are tabs for 'Job Maint.', 'Dependencies', 'Job Input', 'Reports', and 'Miscellaneous'. The main area is divided into two sections: 'Job selection' and 'Job Properties'.

**Job selection**

Region	Job Name	Description
JOBS2	CHQADD	UNPAID CHECK LISTING, BY QUEUE
JOBS2	TODDBT1	DAILY / NIGHTLY PROCESSING JOB #1

Buttons: New, Change, Remove, Copy, Refresh

Class filter: \* Region filter: JOBS2

**Job Properties**

Job Name: JCL  
Region: [dropdown]  
Description: [text area]

Security Req.

Security Key(s)  1  2  3  4  5  6  
 7  8  9  10  11  12  
 13  14  15  16  17  18  
 19  20  21  22  23  24

Class restrict: [text area]

Buttons: Save, Undo, Cancel, Exit

Footer: 0.8 Sec | JOBS2 | JMAINT | Enhanced | 002/000 | Inactive | a # [up] [down] [lock]

JOB Administrator function – Initial View

The above figure shows the initial view of the Administrator screen. Note that the lower portions of the display are all disabled, leaving only the Job Selection list box, the Class and Region filters and the Refresh button enabled. Also note that the overall display area is created as a 'tabbed', notebook display which allows quick and easy access to the entire collection of job settings by simply clicking the desired tab.

Initially the Class filter will be set to an asterisk, indicating all classes for which the current user's security will allow. The region filter is initially set to the region the current session is attached to. The drop down lists for both filters will include all those values that the current users security allows.

The contents of the Job Selection list will include all those jobs that the current Class and Region filters allow. Note that the Refresh button must be clicked when changes are made to either filter value.

To add a new job definition the New button may be clicked at anytime, providing a change to an existing entry is not already underway. In such a case, the New button, as well as all other buttons in the Job Selection are disabled as shown in the next figure, below.

To change an existing job, locate the job's entry in the Job Selection list box and highlight it by clicking it with the mouse. Once selected the Change button will be enabled. Click the Change button to access the jobs information across this and the other tabs shown.

To delete an existing job, select the job's entry in the list as though you were about to change it. The Remove button will also be enabled at this point. Clicking the Remove button will delete the jobs definition from the system. NOTE, be sure the job being deleted will no longer be required as there is no 'un-delete' capability provided. Also note that any existing entries in the System Job Queue that reference this job will remain intact as each Job Queue entry contains it's own unique copy of it's jobs definitions. Removal of job definitions here will not affect the performance or ability of any such already submitted job.

To copy an existing job definition from one region to another, again highlight the job as though it were to be changed. Once highlighted, click the Copy button which will display a dialog box containing a list of region names to which the job definition may be copied.

The individual items that may be configured for a given job are detailed in the next sections. It is important to understand however, that many of the settings made on the job definition screens are used as defaults when the particular job is being selected and submitted to Eden Server for execution. The documentation for each setting that can be made here will indicate whether or not it may be overridden at jobs submission time.

## Job Maintenance Tab

When a new job is being added, or an existing job is being changed, the lower portion of the Job Maintenance tab will be enabled, allowing items to be changed or added. An example of a change in progress is shown below.

The screenshot shows a dialog box titled "Job Properties" with the following fields and controls:

- Job Name: TODDBT1 JCL Q:\SHSA\JCLDOS\TODDBT1.BAT
- Region: JOBS2 (dropdown menu)
- Description: DAILY / NIGHTLY PROCESSING JOB #1 (text input)
- Security Req.:
- Security Key(s):  1,  2,  3,  4,  5,  6,  7,  8,  9,  10,  11,  12,  13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24
- Class restrict: (empty text input)

Buttons at the bottom: Save, Undo, Cancel, Exit

### Job Properties change

The Job Properties area as it appears during a change of an existing job is shown above. Note that the Job Name and Region fields are both protected and cannot be changed for existing jobs. During add of a new job, these fields are enabled and may be changed.

The Description field is used throughout the JOBS administrator and JOBS user interfaces where ever appropriate. Enter a brief description, up to 40 characters, of the job.

The Security settings on this tab, are used to control access during jobs submission by a user as well as when a user accesses any Job Queue entries which reference this job. To enable security checking for this job, the Security Required check box must be selected. Note that when this check box is unselected the individual Security Key check boxes will be disabled, indicating that any user regardless of security may access this job. If Security is required for this job, simply select the Security Required check box and then select at least one Security key check box. When active, security is enforced by requiring the users security settings defined in EManager to match the requirements for the job. If the user has at least one matching key value between their EManager defined settings and the jobs settings they will be able to access the job as it will appear in JOBS systems selection lists and displayed lists of jobs .

Further security may also be defined by using the Class Restrict entry field. When class values are entered in the Class Restrict field, these values are compared to the users EManager defined 'allowed class' values and the job will either be included or excluded from whatever processing the user is attempting. Additionally, use of the Class Restrict field will limit which Eden Server PID's the job will be allowed to run in when it is submitted from the JOBS application. If no class values are entered, there will be no restrictions placed on this job that are class based in terms of which Eden PID's it can run in or in which users can access it. Note that valid class values are any alphabetic character, A through Z.

## Dependencies Tab

The Dependencies tab, shown in the figure below, provides the ability to define run time dependencies for the job in question.

The screenshot shows the 'Edens Client - 0001' window with the 'Dependencies' tab selected. The interface is divided into several sections:

- Job Selection:** Shows 'CTG TEST1 TEST JOB 1'.
- Pre-Run requirements:** A text area containing 'TEST1 must have ended with an RC = 000000 on its last run'. Below it are 'New', 'Change', and 'Remove' buttons. A form below this section includes fields for 'Job name', 'Condition', 'Event', 'Max RC', 'Max Latency', and 'Retry Cnt', along with 'Dates', 'Save', and 'Cancel' buttons.
- Post-Run requirements:** A text area containing 'Run TEST2 as a new PID based job when RC = 000000'. Below it are 'New', 'Change', and 'Remove' buttons. A form below this section includes fields for 'Max RC', 'Job name', 'As new job' (checked), and 'As async PID' (unchecked), along with 'Dates', 'Save', and 'Cancel' buttons.
- Post run dependency entry added:** A confirmation message with 'Save', 'Undo', 'Cancel', and 'Exit' buttons.
- Status Bar:** Shows '0.3 Sec', 'JOBS2', 'DEPEND', 'Enhanced', '014/000', 'Inactive', and a user icon.

As shown in the above figure, there are two types of dependencies supported by JOBS; pre-run and post-run.

Pre-run dependencies are used to define events that must have (or may not have) occurred prior to the start time of the job in question. Post-run dependencies are used to define actions that Eden Server should take upon completion of the job in question.

## Pre-Run Dependencies

Pre-Run dependencies provide the ability to have the running of a job be dependent upon the status of a previously run job, as well as the ability to prevent concurrent executions of the same job.

Using the Pre-run selection list, and the New, Change and Remove buttons, the list of specific requirements may be maintained as needed.

To add a new item, simply click the New button, which is enabled at all times unless a change or add of a pre-run item is already in progress (as is shown in the above figure). Up to 100 pre-run dependencies may be added to a job.

To change an existing item, or to remove one, simply highlight the item with the mouse pointer and click the appropriate button.

After using either the New or the Change button, the entry items in the Pre-Run requirements area are enabled and may be set to their required values. The items that define a pre-run requirement, and their meanings and uses are as follows:

**Job Name** defines the name of the job which will be checked prior to starting this job. The Job name drop down list will include all jobs for the same region as the job being defined that the user also has security access to.

**Condition** defines the logical comparison that will be performed by the Job Scheduler. Possible settings for condition are 'May not have' and 'Must have'. In either case, the condition chosen relates to the Event and other settings for the pre-run requirement.

**Event** defines the type of event that the job scheduler will be attempting to locate when processing this pre-run requirement. The event can be set to be either 'Completed', 'Started', 'Abended' or 'Run Concurrently'. **Completed** events are satisfied by locating a job queue entry for the indicated pre-run job name that has completed, regardless of the completion status. That is, Completed jobs include any job that started and then ended, including jobs that abended. **Started** jobs include any jobs that was started, including those jobs that are completed, abended and or still executing. Abended jobs include only those jobs which ended abnormally due to a JCL or other internal job processing error. Note that jobs that ended abnormally due to, for example, a file open error in a COBOL program are not considered to have abended. Abended jobs will also always have a maximum/final return code of 34034.

**Max RC** defines the pre-run job's maximum/final return code and the type of comparison to that return code that the job scheduler should perform. When entering a Max RC value, the format of the entry should be: CC nnnnnn where CC is a comparison code and nnnnnn is the return code value to compare against. The possible values for CC are:

EQ, meaning 'equal'

GT, meaning 'Greater Than'

LT, meaning 'Less Than'

GE, meaning 'Great Than or Equal' (note: a Max RC setting of EQ 000000 effectively means any/all return codes, as a job will always have an RC value of at least zero).

LE, meaning 'Less Than or Equal'

NE, meaning 'Not Equal'

**Max Latency** provides the ability to define a time frame during which the event in question must have occurred, or in the case of 'May not have' type of conditions that the event did not occur within a time period. For example, the pre-run example shown in the figure above includes a nineteen hour period during which the CHQADD job may not have completed at any time during the previous 19 hours. If a time is specified it must be entered in standard 24 hour notation in the HH:MM:SS format. The actual wall clock time for the period in question is determined by the job scheduler by counting backwards from the current time (meaning the base job's start time). If 00:00:00 is entered for a latency period, the job scheduler interprets this to mean the last time the pre-run job completed/abended/started, etc without regard to any specific time period.

**Retry Count** is used only in conjunction with 'Run concurrently' events. When specified, the retry count will be used as a count of 5 minute intervals during which the job scheduler will attempt to satisfy a 'may not run concurrently' event. For example, suppose job 'TEST1' has a 'may not run concurrently' pre-run requirement that specifies job 'TEST2' as it's pre-run job name. Now, if TEST1 were scheduled to start and at the appointed start time job TEST2 is running, the job scheduler will retry starting TEST1 every 5 minutes until the retry count has been exhausted. Leaving the retry count field blank, or entering zero, will cause the pre-run requirement to fail in such a case.

Once settings for the pre-run requirement are complete, the Save button may be clicked. When clicked, the pre-run requirement is edited and then displayed as an easy-to-read sentence as shown in the above figure.

A maximum of 100 pre-run requirements may be defined for any given job. If for any reason a pre-run requirement fails, the job will not actually be executed, however a job log will be created and the job will be marked as 'completed'.

Note that by using the email features (see the Miscellaneous tab) it is possible to have Eden Server automatically notify a list of recipients in the event of a pre-run failure.

## Post-Run Dependencies

Post-Run dependencies provide a method of triggering the execution of other jobs when the current job completes.

Using the Post-run selection list, and the New, Change and Remove buttons, the list of specific requirements may be maintained as needed.

To add a new item, simply click the New button, which is enabled at all times unless a change or add of a post-run item is already in progress (as is shown in the above figure). Up to 100 post-run dependencies may be added to a job.

To change an existing item, or to remove one, simply highlight the item with the mouse pointer and click the appropriate button.

After using either the New or the Change button, the entry items in the Post-Run requirements area are enabled and may be set to their required values. The items that define a post-run requirement, and their meanings and uses are as follows:

**Max RC** defines the post-run job's maximum/final return code and the type of comparison to that return code that the job scheduler should perform. When entering a Max RC value, the format of the entry should be: CC nnnnnn where CC is a comparison code and nnnnnn is the return code value to compare against. The possible values for CC are:

EQ, meaning 'equal'

GT, meaning 'Greater Than'

LT, meaning 'Less Than'

GE, meaning 'Great Than or Equal' (note: a Max RC setting of EQ 000000 effectively means any/all return codes, as a job will always have an RC value of at least zero).

LE, meaning 'Less Than or Equal'

NE, meaning 'Not Equal'

**Job Name** defines the name of the job which will be (conditionally) started upon completion of the current job. The Job name drop down list will include all jobs for the same region as the job being defined that the user also has security access to.

The **As new job** radio button specifies that the job to be run should be started as a normal job; that is, that it should acquire its own PID and execute separately from the current job.

The **As async PID** radio button specifies that the job to be run should be loaded and executed within the context of the current job, however it should be run in an async PID. Async PID's are described in detail in the programmers reference – see the JCL section referencing the ASYNC command. Briefly, however, an Async PID may be thought of in the same manner as a CALL statement in a .bat file, except that control returns to the .bat issuing the CALL prior to completion of the .bat being CALL'ed.

Use of the Async option here is recommended for situations where multiple post-run

jobs must be run and all such post run jobs may run concurrently. For example, consider the situation where multiple data files must be restored if a job ends with an RC indicating failure. By running multiple file restore jobs concurrently, the overall duration of the restore process can be significantly reduced.

Once settings for the post-run requirement are complete, the Save button may be clicked. When clicked, the post-run requirement is edited and then displayed as an easy-to-read sentence as shown in the above figure.

A maximum of 100 post-run requirements may be defined for any given job. If for any reason a post-run requirement is unable to be started it is possible to have Eden Server automatically notify a list of recipients of the failed post run event. See the Miscellaneous tab for information on specifying email notifications.

## Dependency Date Filters

In addition to the base capabilities pre and post run dependency provide, each of these types of dependencies may include Date Filters. Through the use of date filters a single job may be set up in such a way as to support complex scheduling requirements.

For example, suppose that a nightly run consists of a series of jobs that always starts off by running a system backup, however, the jobs executed after the backup vary depending upon the date or day of the week. A good example is 'month end' type processing, where the nightly schedule includes all those jobs that typically run every day, but on the last day of the month an additional series of job must be run. Eden Server supports this type of complex scheduling through the use of Date Filters applied to the pre- and post-run dependencies for the jobs in question.

Typically a dependency, either pre or post, is processed by the scheduler based solely on the settings of the dependency itself, for example job 'a' is run when job 'a' completes with a return code of 4 or less. Including a date filter for a dependency allows Eden Server to process or ignore a given dependency based on the date filter in comparison to the current run date.

When date filters are present for a dependency, Eden first analyzes the date filters and then determines if the current run date matches the filter. If the run date matches the filter, then the dependency is processed. If the run date does not match the filter then the dependency is ignored and Eden Server continues processing as though the dependency were not present.

The items that may be configured for a date filter are shown in the figure below.

**Dependency date filters**

Months		Days and Weeks		Date
<input checked="" type="checkbox"/> Jan	<input type="checkbox"/> Jul	<input type="checkbox"/> Mon	<input checked="" type="radio"/> 1st Full	Day of Month <input type="text"/>
<input type="checkbox"/> Feb	<input type="checkbox"/> Aug	<input type="checkbox"/> Tue	<input checked="" type="radio"/> 1st Day	Skip Holidays <input type="checkbox"/>
<input type="checkbox"/> Mar	<input type="checkbox"/> Sep	<input type="checkbox"/> Wed	<input type="checkbox"/> 1st week	<b>Filter type</b>
<input type="checkbox"/> Apr	<input type="checkbox"/> Oct	<input type="checkbox"/> Thu	<input type="checkbox"/> 2nd week	<input type="text" value="Matched"/>
<input type="checkbox"/> May	<input type="checkbox"/> Nov	<input type="checkbox"/> Fri	<input type="checkbox"/> 3rd week	
<input type="checkbox"/> Jun	<input type="checkbox"/> Dec	<input type="checkbox"/> Sat	<input type="checkbox"/> 4th week	
		<input type="checkbox"/> Sun	<input type="checkbox"/> Last week	

Reset Ok Cancel

The date filters dialog, shown above, is divided into four areas: The Filter Type, Months, Days and Weeks and Date areas. By combining various entries in these areas, complex filters may be built that may be used to describe nearly any type of scheduling requirements.

The controls displayed in each of these areas, and their uses are as follows:

The **Filter Type** drop down list describes what sort of comparison Eden Server should make when comparing the current run date to the date filter. The two possible choices are 'Matched' and 'Not Matched'. A setting of 'Matched' indicates that date filtering will 'pass' when the current run date matches the filter, thus causing the dependency to be processed. A setting of 'Not Matched' indicates that the date filtering will only 'pass' when the current run date does *not* match the date filter. Therefore, use of the Matched / Not Matched setting can be used to create two filters that will cause one of two outcomes. For example, creating a post-run dependency to run job 'a' on the last day of the month and run job 'b' on all other days of the month requires two 'last day of the month' filters – the first dependency is for job 'a' and should be 'Matched', the second dependency is for job 'b' and should be 'Not Matched'.

The **Months** area allows individual months to be selected as part of the filter. If a month is selected, the filter will apply only to that month. Note that multiple months may be selected. Also note that selecting all 12 months is the same as selecting no months. This is due to the fact that leaving all months un-selected causes Eden Server to disregard what month it is, which is the same as selecting all 12 months.

The **Days and Weeks** area allows individual days of the week and, optionally, a week of the month to be included as part of the filter. Selecting a specific day, or days, the filter can be made to apply only to those days of the week. For example, if a post-run dependency should only be processed on week days, selecting the 'Mon' through the 'Fri' check boxes will cause the dependency to be processed on Monday through Friday only.

Adding to the Days options are the Weeks options. If a dependency is only to be processed on a particular day of the week, during a particular week of the month, the week of the month check box for that week (or weeks) may be selected. Note that selecting a Weeks option is only allowed if a Days option is also selected. Also, note that when selecting a weeks option, selection of a 'weeks method' is also required. Eden Server allows for two methods when determining what 'week of the month' a date falls in. The first method is '1<sup>st</sup> Full' which causes Eden to consider the first week of the month to be the first full week. In this case, Eden counts weeks starting on Monday's. Therefore, the first week of the month starts on the first Monday of the month. The second method is the '1<sup>st</sup> day' method which allows Eden to start counting weeks on the first day of the month. In this case, if the first of the month were a Tuesday, selecting the first Monday during the 1<sup>st</sup> week would result in selecting Monday the 7<sup>th</sup>.

The **Day of Month** field allows for the entry of a specific date during the month, for example the 1<sup>st</sup> or 15<sup>th</sup> of the month. Additionally, the figurative constant 'LAST' may also be entered here to indicate the last day of the month. Note that use of the Day of

the Month option may not be combined when using a Day of the week option. Use of the Day of Month option may, though, be combined with the Months options. For example, to create a dependency that is effective only on the last day of each quarter, select the 4 months during which the quarters end and also enter LAST in the Day of Month.

The **Skip Holidays** option works in conjunction with the Eden Server holiday schedule which allows for the definition of business holiday dates to be defined. By selecting the Skip Holidays check box, filters may be made to be 'holiday sensitive'.

The **Reset** button causes all items in the date filter to be reset to 'off', effectively removing the date filter from the dependency. To remove a date filter from a dependency use this button, then **Ok**, then save the dependency.

The **Ok** button causes the date filter to be edited for validity, and if no errors are found, the filter dialog is removed allowing further edits to the dependency. Note, the Ok button does not save the date filter, the Save button in the pre or post dependency area must still be used to save the date filters to the dependency in question.

The **Cancel** button removes the filter dialog display and causes any changes made to the filter to be ignored.

## Job Input Tab

The Job Input tab provides the ability to specify what, if any, user supplied input items are required prior to submitting the job for execution. A sample of the job input tab is shown in the figure below.

Eden Client - 0001

Job Maint. | Dependencies | Job Input | Reports | Miscellaneous

**Job Selection**  
JOBS2 TODDBT1 DAILY / NIGHTLY PROCESSING JOB #1

**Input Parameters**  
Cmdline parm %1 is an optional 'MMDDYY' date

New | Change | Remove

Cmd Line Parm  Parm no %1 Ctl Card Parm   
Ctl Card DD  Step Name   
Entry Required  Ctl Card Start   
Parm type MMDDYY date Max Length 006  
Field Label TEST F  
User Help Text Enter a date in the format of MMDDYY, without slashes. This date controls the time period of the report produced.

Save | Cancel

Save | Undo | Cancel | Exit

0.3 Sec | JOBS2 | INPUT | Enhanced | 011/016 | Inactive | a # ↕ 🔒

Sample Job input tab display

The Eden Server JOBS system supports user input items for batch jobs in two modes; Command Line parameters and Control Card input.

From the users viewpoint, at job submission time, there is no difference between these

two types of input items, however several differences exist both during the definition stage and during actual job execution. Command line parameters are, as would be expected, passed to the job in question via the starting command line. Subsequently, these items may then be accessed within the job's .bat file through the use of %1 through %9 replaceable variables.

Control Card parameters are processed at job execution time by writing the user supplied text to the indicated control card disk file. Note that when control card input items are used, the Eden Server background manager will create or modify the required control card as needed. If a job could potentially be submitted for concurrent execution by different users it is therefore recommended that all control cards that may be modified by the background manager include some portion of their name that will be unique to the particular instance of each job. For example, the easiest way to ensure unique disk file names is to suffix the files disk name with the %jobno% Eden Server variable. See the Eden Programmers Reference for complete information on this Eden supplied variable.

The entry items required to define an input item are as follows:

The **Cmd Line Parm** radio button should be selected if the item in question is to be passed to the job as a command line parameter. Note that selection of this option will require an entry in the 'Parm no' field. Also, this choice will cause the Ctl Card DD, Step Name and Ctl Card Start items to be disabled as these items are not relevant for command line parameter items.

The **Parm no** field defines which command line parameter, %1 through %9 this input item will occupy. Note that this item will be disabled if the Ctl Card Parm option is selected.

The **Ctl Card Parm** radio button should be selected if the item in question is to be passed to the job as text written to a control card disk file. Note that selection of this option will require entries to be made in the Ctl Card DD, Step Name and Ctl Card Start fields. Also, if this option is selected the Parm No field will be disabled.

**Ctl Card DD** is used to specify the DD name of the control card which should be updated with this item's input text. Note that the 'DD name' of the card in question must be defined via a .bat file SET statement. For example, SET ctl-card-dd=disk-file-name. Note that this field is only enabled when the Ctl Card Parm choice is selected.

**Step Name** is used to specify which step name, within the overall .bat file, contains the definition of the control card in question. Note, as the specific control that is updated is defined through a combination of Ctl Card DD and Step Name, it is possible to have the same control card DD name specified in multiple steps of a job. Note that this field is only enabled when the Ctl Card Parm choice is selected.

The **Entry Required** check box is used by the job submission function to determine if it is permissible to submit a job run request without all input items being completed.

The **Ctl Card Start** field is used to define which column within the control card the text data supplied by the user should be written to. Note that this field is only enabled when the Ctl Card Parm choice is selected. The value entered may be any column number from 1 (the first column in a card) through 80 (the last available column position).

The **Parm Type** selection list contains the allowable contents of the input item. The selection list contains all standard date formats as well as several free-form type formats which may be used if no other specific type is suitable for a particular item.

The **Max Length** field is used to define the total number of characters to be entered by the user at job submission time.

The **Field Label** field may be completed with a brief (15 character maximum) 'label' which will be displayed adjacent to the actual input field on the job submission window.

The **User Help** Text field may be completed with up to 144 characters of optional user help text. If present, the supplied text will be used by the job submission window as context sensitive help that the user may display by using Eden Clients context sensitive help feature.

When all fields pertaining to the input item are complete the **Save** button may be used to record the item and save it to the list of other input items. Note that even after saving the item(s) it is still required that the overall job definition record be saved by using the **Save** button located at the lower left corner of this and all other JOBS administrator tabs.

Clicking the **Cancel** button will stop the add or change process and leave the existing items intact.

## Reports Tab

The reports tab is used to define the output print reports produced by the job as well as to define retention and other information specific to each report. An example of a reports tab display is shown in the figure below.

Eden Client - 0001

Job Maint. | Dependencies | Job Input | **Reports** | Miscellaneous

Job Selection  
JOBS2 CHQADD UNPAID CHECK LISTING, BY QUEUE

Report List

Base DSN	Description	Max days	Security
CQARPT	ADDUP REPORT	0	Required

New | Change | Remove | Scan JCL

Base DSN: CQARPT  
Description: ADDUP REPORT  
Max Save Days: 0  
Security Req.   
Security Key(s)  1  2  3  4  5  6  
 7  8  9  10  11  12  
 13  14  15  16  17  18  
 19  20  21  22  23  24

Save | Cancel

Save | Undo | Cancel | Exit

0.4 Sec | JOBS2 | REPORT | 011/016 | Inactive | a # ↕ 🔒

Sample reports tab display

The Reports tab provides the ability to define up to 40 output reports and their associated information.

To add a new report description, either click the New button or the Scan JCL button,

which are shown at the left and right ends of the row of buttons under the report selection list.

Clicking the New button will allow free-form entry of a new report. Clicking the Scan JCL button causes the JOBS administrator to scan the JCL member used for this job and to analyze the SETQ statements found therein. Note that the Scan JCL button is the recommended method for adding reports as it's use removes much of the possible user-error that could occur through either omitting a report, or by making a typographical error in the reports definition.

To change or remove an existing entry, simply highlight the item in the report selection list and click the appropriate button.

The items required for report definitions are as follows:

**Base DSN** defines the base name of the file as specified in the SETQ statement in the JCL member. The Base DSN is derived from the item specified in the SETQ as follows: SETQ DD-name=Base-DSN,....

**Description** is a brief (up to 40 characters) description of the report and may contain any text.

**Max Save Days** defines the maximum number of days the report will be maintained on the Eden Server print queue. Note that if the report is to be maintained indefinitely, a value of 0 (zero) days may be specified.

**Security Required** controls whether or not access to this report requires the users login privileges to include at least one security key value that matches one of the security keys selected for this report.

**Security Keys** specifies, in the event that security key based access is required, which particular security keys a users sign on privileges must include before access is granted to the report. Note that the use of security key access control is in addition to any other security checks which may be in place, such as class or user-id based access control.

Once all items for the report definition have been completed, the Save button should be clicked to update the report selection list. Note that even after this save button is clicked, the overall job definition must be saved by clicking the Save button located at the lower left corner of this, and all JOBS administrator tabs.

## Miscellaneous tab

The Miscellaneous tab provides the ability to specify both email notifications as well as to include help text for the job which may be accessed by the user at job submission time through using Eden Client's context sensitive help feature.

The screenshot shows the 'Miscellaneous' tab in the Eden Client application. The window title is 'Eden Client - 0001'. The interface includes a menu bar with 'Job Maint.', 'Dependencies', 'Job Input', 'Reports', and 'Miscellaneous'. The main content area is divided into several sections:

- Job selection:** A text box containing 'SHSA TODDBT1 DAILY BATCH JOB NUMBER 1'.
- Email Notifications:** Three text boxes for 'Pre-run Failure', 'Job Abend', and 'Post-run Failure', all containing 'ETHANS@ROSEBUDUSA.COM'. A tooltip points to the 'Pre-run Failure' field with the text 'Email recipient list for Pre-Run failure notifications'.
- Job Help Text:** A large empty text area.
- Job Flow Control:** A section with a 'Job Enabled' checkbox (checked), a 'Hold Response' dropdown set to 'COMPRESPOND', and radio buttons for 'Pre Run Hold' (unselected) and 'Post Run Hold' (selected).

At the bottom of the main area are buttons for 'Save', 'Undo', 'Cancel', and 'Exit'. The status bar at the very bottom shows '0.5 Sec', 'SHSA', 'MISC', '019/028', 'Inactive', and a set of icons.

Sample Miscellaneous tab display

The entry items on this tab are as follows:

The **Pre-Run Failure** email address list allows for the entry of multiple email addresses. In the event of a pre-run failure of any sort, Eden Server will send an email notification detailing the failure and the exact reason for it to each recipient listed here. Note that when entering multiple recipients their email addresses must be separated by either a space or a comma.

The **Job Abend** email address list allows for the entry of multiple email addresses. In the event this job abends due to either a JCL error or other internal processing error, Eden Server will send an email notification detailing the abend and the exact reason for it to each recipient listed here. Note that when entering multiple recipients their email addresses must be separated by either a space or a comma.

The **Post-run failure** email address list allows for the entry of multiple email addresses. In the event any of this jobs post-run requirements are unable to be started, Eden Server will send an email notification detailing the failure and the exact reason for it to each recipient listed here. Note that when entering multiple recipients their email addresses must be separated by either a space or a comma. Also, note that a 'post-run failure' does not include post-run requirements that are ignored due to their return code checking indicating they are not needed. Instead, a post-run failure is only indicated by an internal processing error, such as the post-run job's JCL member being not-found or some other similar type of error.

The **Job Help** field provides for entry of context sensitive help text that may be displayed by the user at job submission time. To access this help text, the user must already have selected the job for submission. The help text entered here will be displayed when the user selects the context sensitive help button from the Eden Client toolbar and then click on any open space on the job submission display tab.

The **Job Enabled** check box sets the processing status for this job. When selected, the job will execute as expected whenever the Eden Scheduler receives a run request for the job. When the check box is not selected, the Eden Scheduler will bypass actual JCL execution and instead simply mark the job as 'complete', with a final return code of zero. When the job is not enabled, processing for any defined post run dependencies will take place, however no pre-run processing will be enforced.

The **Hold Response** entry field and the Pre and Post Run Hold buttons may be used to cause the scheduler to either delay processing (pre-run hold) or delay marking the job as finished (post run hold) until the console operator enters the Response Word. This feature is intended to provide the ability to review a particular job's output / results prior to allowing subsequent processing to occur. Note that if a response word is required for a job, the console will display message 0035-999-A, prompting the operator for the Response Word. In the event the job should not proceed, the console STOP *pid-no* command may be entered to cause the job to be cancelled. If the job should be allowed to proceed, the entry of the Response Word will automatically re-activate processing of the job.

## Run Schedule reporting

Eden Server includes a batch utility program called RMSSCHED that may be used to produce run schedule reports detailing proposed run schedules for varying time periods.

Prior to using RMSSCHED, however, it must be set up using the JOBS administrator functions. To start, please review the supplied JCL named RMSSCHED.BAT which is located in the Eden Server install directory. Any changes made to this JCL should be noted and applied to the set up instructions provided herein.

Once any JCL changes have been made to RMSSCHED.BAT, a Job Definition must be created as for any other batch job.

When creating a job definition for RMSSCHED the only required items relate to the control card input that the RMSSCHED.EXE program uses. Note that the control card for RMSSCHED is built as follows:

Columns 1 through 10 contain the 'from date' for the report. The date is a required field and must be in the format MM/DD/CCYY, including the slashes.

Columns 11 through 20 contain the 'thru date' for the report. The date is a required field and must be in the format MM/DD/CCYY, including the slashes.

Column 21 may contain a Y or N value indicating whether the report should include full text descriptions of post run dependencies. Note, this control card item is optional, with the default value being an N.

The following screen shots show the proper definitions to make on the Job Input tab for RMSSCHED:

**RMSSCHED parameter 1**

Parm type and specifications					
Cmd Line Parm	<input checked="" type="radio"/>	Ctlcd Parm	<input checked="" type="radio"/>	Env. Var	<input type="radio"/>
Parm no	<input type="text"/>	Ctlcd DD	<input type="text" value="SYSIN"/>	Variable	<input type="text"/>
		Step Name	<input type="text" value="STEP01"/>		
		Ctlcd Column	<input type="text" value="001"/>		

Common settings			
Parm type	<input type="text" value="MM/DD/CCYY date"/>	Max Length	<input type="text" value="010"/>
Field Label	<input type="text" value="From Date"/>	Entry Required	<input checked="" type="checkbox"/>
User Help Text	<input type="text" value="Enter a from date, in MM/DD/CCYY format, that corresponds to the earliest date to include in the report."/>		
Default value	<input type="text"/>	Save	Cancel

## RMSSCHED parameter 2

Parm type and specifications					
Cmd Line Parm	<input type="radio"/>	Ctlcd Parm	<input checked="" type="radio"/>	Env. Var	<input type="radio"/>
Parm no	<input type="text"/>	Ctlcd DD	<input type="text" value="SYSIN"/>	variable	<input type="text"/>
		Step Name	<input type="text" value="STEP01"/>		
		Ctlcd column	<input type="text" value="011"/>		
Common settings					
Parm type	<input type="text" value="MM/DD/CCYY date"/>	Max Length	<input type="text" value="010"/>		
Field Label	<input type="text" value="To Date"/>	Entry Required	<input checked="" type="checkbox"/>		
User Help Text	<input type="text" value="Enter a from date, in MM/DD/CCYY format, that corresponds to the latest date to include in the report."/>				
Default value	<input type="text"/>	Save	Cancel		

## RMSSCHED parameter 3

Parm type and specifications					
Cmd Line Parm	<input type="radio"/>	Ctlcd Parm	<input checked="" type="radio"/>	Env. Var	<input type="radio"/>
Parm no	<input type="text"/>	Ctlcd DD	<input type="text" value="SYSIN"/>	variable	<input type="text"/>
		Step Name	<input type="text" value="STEP01"/>		
		Ctlcd column	<input type="text" value="021"/>		
Common settings					
Parm type	<input type="text" value="'Y' or 'N' value"/>	Max Length	<input type="text" value="001"/>		
Field Label	<input type="text" value="Show Dependency"/>	Entry Required	<input type="checkbox"/>		
User Help Text	<input type="text" value="Enter a 'Y' to include text explanations of all post-run dependencies, date filters as well as the post-run results."/>				
Default value	<input type="text"/>	Save	Cancel		

All other configuration items for RMSSCHED are optional.